



3 1176 00159 6049

NASA CONTRACTOR REPORT 166316

*NASA CR-166,316*

NASA-CR-166316  
19820014269

A Development of Grid Procedure for  
Multicomponent Aerodynamic Configuration

H.C. Chen

**FOR REFERENCE**

NOT TO BE TAKEN FROM THIS ROOM

**LIBRARY COPY**

APR 22 1982

LANGLEY RESEARCH CENTER  
LIBRARY, NASA  
HAMPTON, VIRGINIA

CONTRACT NAS2-10676  
November 1981

**NASA**



NF02305



NASA CONTRACTOR REPORT 166316

A Development of Grid Procedure for  
Multicomponent Aerodynamic Configuration

H.C. Chen  
The Boeing Company  
P.O. Box 3707  
Seattle, Washington 98124

Prepared for  
Ames Research Center  
under Contract NAS2-10676



National Aeronautics and  
Space Administration

Ames Research Center  
Moffett Field, California 94035

N82-22143 #

A DEVELOPMENT OF GRID GENERATION PROCEDURE FOR  
MULTICOMPONENT AERODYNAMIC CONFIGURATION

H. C. Chen

The Boeing Company  
Seattle Washington 98124

Prepared for ARC Under Contract NAS2-10676

SUMMARY

An extension of the approximation factorization (AF) scheme to multi-block grids is studied. An innovative method involving zonal decomposition using interfacing grids is investigated. Suitably matching the solutions at a block boundary for an incompressible flow problem it is demonstrated that this method is feasible for both contiguous and noncontiguous interfacing grids. Remaining issues for extending this method to transonic flow problem are discussed. The flow solution algorithm of a fully implicit AF code (TWINGB) is investigated for adaptation to block-structured grids. The analysis shows that the TWINGB code is sensitive to grid effects. A method to improve the TWINGB flow solution algorithm is presented to reduce the grid dependency of the code. Test case studies for subsonic and transonic flow over parabolic arc airfoils and wings suggest that the improved method should be adapted for extending TWINGB more effectively to flow problem involving arbitrary grids with block structure.

# TABLE OF CONTENTS

	Page
I. INTRODUCTION . . . . .	1
II. BLOCK-STRUCTURED GRID . . . . .	5
III. AF2 SCHEME . . . . .	10
A. Governing Equations . . . . .	10
B. Flow Solution Algorithm . . . . .	14
C. AF2 Algorithm . . . . .	17
D. Temporary Damping . . . . .	19
E. Boundary Conditions . . . . .	19
IV. SOLUTION BY ZONAL DECOMPOSITION . . . . .	20
A. Model Problem . . . . .	21
B. Flow Field Matching Procedure . . . . .	23
C. Calculation of $\phi_n$ along EF . . . . .	23
D. AF Scheme . . . . .	24
E. Baseline Solutions . . . . .	25
F. Flow Field Matching with Contiguous Interfacing Grids . . . . .	26
G. Flow Field Matching with Noncontiguous Interfacing Grids . . . . .	29
H. Conclusions with Zonal Decomposition . . . . .	32
I. Remaining Issues . . . . .	33
V. GRID EFFECT STUDY ON TWINGB CODE . . . . .	34
A. Density Calculation in TWINGB . . . . .	34
B. Effect Due to Uneven Grid Spacing . . . . .	37
C. Effect Due to Grid Distortion . . . . .	38
D. Effect Due to Grid Aspect Ratio . . . . .	44
E. Conclusions on Grid Effect Study . . . . .	44
VI. RESULTS FROM ADAPTATION OF TWINGB TO BLOCK-STRUCTURED GRIDS . . . . .	51
A. Test Cases for Airfoil Problems . . . . .	52
B. Test Cases for Wing Problems . . . . .	72
C. Test Case Study Summary . . . . .	94
VII. CONCLUSIONS . . . . .	97
VIII. RECOMMENDATIONS . . . . .	98
IX. REFERENCES . . . . .	99
X. APPENDICES . . . . .	100

## LIST OF FIGURES

- Figure 1. Block Structuring.
- Figure 2. A Typical Block-Structured Grid.
- Figure 3. Comparison of Grids Near an Airfoil.
- Figure 4. Comparison of Grid Structure.
- Figure 5. Block-Structured Grid Generation Process.
- Figure 6. Alternating Boundary Conditions Flow Field Matching.
- Figure 7. Convergence Histories on Coarse Mesh.
- Figure 8. Convergence Histories on Fine Mesh.
- Figure 9. Flow Field Matching With Noncontiguous interfacing Grids.
- Figure 10. Grid Effect on  $\phi_x^2$  Value Computed in TWINGB.
- Figure 11. A Simple Grid to Test The Grid Sensitivity of TWINGB.
- Figure 12. Error in Density Due to Grid Effect.
- Figure 13. A Smooth Grid for Circular Cylinder Case,  $FR=1.2$ .
- Figure 14. A Locally Distorted Grid for Circular Cylinder Case,  $FR=1.2$ .
- Figure 15. Grid Effect Study, Circular Cylinder,  $M_\infty=.45$ , Upper Surface M.
- Figure 16. Grid Effect Study, Circular Cylinder,  $M_\infty=.01$ , Upper Surface M.
- Figure 17. A Smooth Grid for Circular Cylinder Case,  $FR=1.5$ .
- Figure 18. Isoparametric Mapping and Subdomain Definition.
- Figure 19. Grid-A, Flat Plate, Uniform Grid.
- Figure 20. Grid-B, Flat Plate,  $FZ=1.2$ .
- Figure 21. Growth of  $R_{max}$  and  $C_{max}$  for Flow Over a Flat Plate, Grid-B,  $M_\infty=.0001$ .
- Figure 22. Grid-C, Flat Plate,  $FZ=1.1$ .
- Figure 23. Grid-D, Flat Plate,  $FX=1.125$ ,  $FZ=1.1$ .
- Figure 24. Convergence Histories for Flow Over a Flat Plate, Grid-D,  $M_\infty=.9$ .

- Figure 25. Convergence Histories for Flow Over a Flat Plate, Grid-D,  $M_\infty = .0001$
- Figure 26. Nonzero  $C_p$  on the Flat Plate Surface Due to Grid Effect, Grid-D.
- Figure 27. Grid-E, .03 Parabolic Arc Airfoil,  $FX=1.125$ ,  $FZ=1.1$ .
- Figure 28. Convergence Histories for Flow Over a .03 Parabolic Arc Airfoil, Grid-E,  $M_\infty = .75$ .
- Figure 29. Surface  $C_p$  for Flow Over a .03 Parabolic Arc Airfoil, Grid-E,  $M_\infty = .75$ .
- Figure 30. Grid-F, .12 Parabolic Arc Airfoil,  $FX=1.125$ ,  $FZ=1.1$ .
- Figure 31. Growth of  $R_{max}$  and  $C_{max}$  for Flow Over a .12 Parabolic Arc Airfoil, Grid-F,  $M_\infty = .8$ .
- Figure 32. Growth of  $R_{max}$  and  $C_{max}$  for Flow Over a .12 Parabolic Arc Airfoil, Grid-F,  $M_\infty = .8$ ,  $\omega=1$ .
- Figure 33. Growth of  $R_{max}$  and  $C_{max}$  for Flow Over a .12 Parabolic Arc Airfoil, Grid-F,  $M_\infty = .8$ ,  $\alpha_L=1$ ,  $\alpha_H=20$ .
- Figure 34. Growth of  $R_{max}$  and  $C_{max}$  for Flow Over a .12 Parabolic Arc Airfoil, Grid-F,  $M_\infty = .8$ ,  $\beta_t = 1$  Everywhere.
- Figure 35. Convergence Histories for Flow Over a .12 Parabolic Arc Airfoil, Grid-F,  $M_\infty = .0001$ .
- Figure 36. Grid-G, .12 Parabolic Arc Airfoil,  $FX=1.125$ ,  $FZ=1$ .
- Figure 37. Growth of  $R_{max}$  and  $C_{max}$  for Flow Over a .12 Parabolic Arc Airfoil, Grid-G,  $M_\infty = .8$ .
- Figure 38. Grid-H, .12 Parabolic Arc Airfoil,  $FX1=1.129$ ,  $FX2=1$ ,  $FZ=1.1$ .
- Figure 39a. Convergence History of  $R_{max}$  for Flow Over a .12 Parabolic Arc Airfoil, Grid-H,  $M_\infty = .8$ .
- Figure 39b. Convergence History of NSUP for Flow Over a .12 Parabolic Arc Airfoil, Grid-H,  $M_\infty = .8$ .
- Figure 40. Surface  $C_p$  Distributions for Flow Over a .12 Parabolic Arc Airfoil, Grid-H,  $M_\infty = .8$ .
- Figure 41. Convergence History for Flow Over a Swept Wing With a .03 Parabolic Arc Wing Section, Planform-A, Grid-E,  $M_\infty = .75$ .
- Figure 42. Grid-I, .03 Parabolic Arc Wing Section,  $FX1=1.129$ ,  $FX2=1$ ,  $FZ=1.1$ .
- Figure 43. Convergence History for Flow Over a Swept Wing With a .03 Parabolic Arc Wing Section, Planform-A, Grid-I,  $M_\infty = .75$ ,  $AR=9.5$ .

- Figure 44. Surface  $C_p$  Distributions for Flow Over a Swept Wing With a .03 Parabolic Arc Wing Section, Planform-A, Grid-I,  $M_\infty = .75$ ,  $AR=9.5$ .
- Figure 45. Convergence History for Flow Over a Swept Wing With a .03 Parabolic Arc Wing Section, Planform-B, Grid-I,  $M_\infty = .75$ ,  $AR=1.9$ .
- Figure 46. Surface  $C_p$  Distributions for Flow Over a Swept Wing With a .03 Parabolic Arc Wing Section, Planform-B, Grid-I,  $M_\infty = .75$ ,  $AR=1.9$ .
- Figure 47a. Convergence History of  $R_{max}$  for Flow Over a Swept Wing With a .12 Parabolic Arc Wing Section, Planform-A, Grid-H,  $M_\infty = .8$ ,  $AR=9.5$ .
- Figure 47b. Convergence History of NSUP for Flow Over a Swept Wing With a .12 Parabolic Arc Wing Section, Planform-A, Grid-H,  $M_\infty = .8$ ,  $AR=9.5$ .
- Figure 48. Shock Wave Position Relative to the Wing Planform, Planform-A,  $AR=9.5$ .
- Figure 49. Surface  $C_p$  Distributions for Flow Over a Swept Wing With a .12 Parabolic Arc Wing Section, Planform-A, Grid-H,  $M_\infty = .8$ ,  $AR=9.5$ .
- Figure 50a. Convergence History of  $R_{max}$  for Flow Over a Swept Wing With a .12 Parabolic Arc Wing Section, Planform-B, Grid-H,  $M_\infty = .8$ ,  $AR=1.9$ .
- Figure 50b. Convergence History of NSUP for Flow Over a Swept Wing With a .12 Parabolic Arc Wing Section, Planform-B, Grid-H,  $M_\infty = .8$ ,  $AR=1.9$ .
- Figure 51. Shock Wave Position Relative to the Wing Planform, Planform-B,  $AR=1.9$ .
- Figure 52. Surface  $C_p$  Distributions for Flow Over a Swept Wing With a .12 Parabolic Arc Wing Section, Planform-B, Grid-H,  $M_\infty = .8$ ,  $AR=1.9$ .



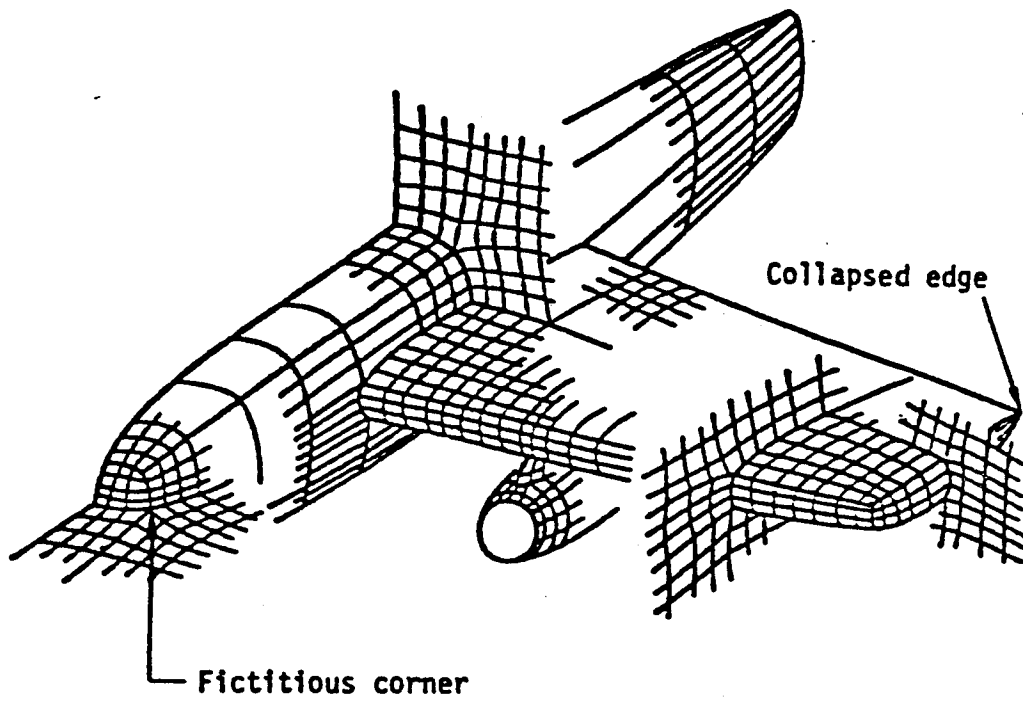
## I. INTRODUCTION

The objective of the present study is to explore means for extending approximate factorization (AF) schemes (reference 1) to the treatment of transonic flows requiring use of complex, multi-block grids (reference 2) such as depicted in Figure 1. It has been demonstrated on a single-block ring grid that the fully implicit nature of the AF scheme (reference 1) significantly improves the convergence speed of the flow solution relative to successive line overrelaxation (SLOR) scheme. It is highly desirable to extend this scheme to block-structured grid. The primary issues involved center about devising means for extending the transonic finite difference algorithms beyond the simplicity inherent in a simple, single-block grid.

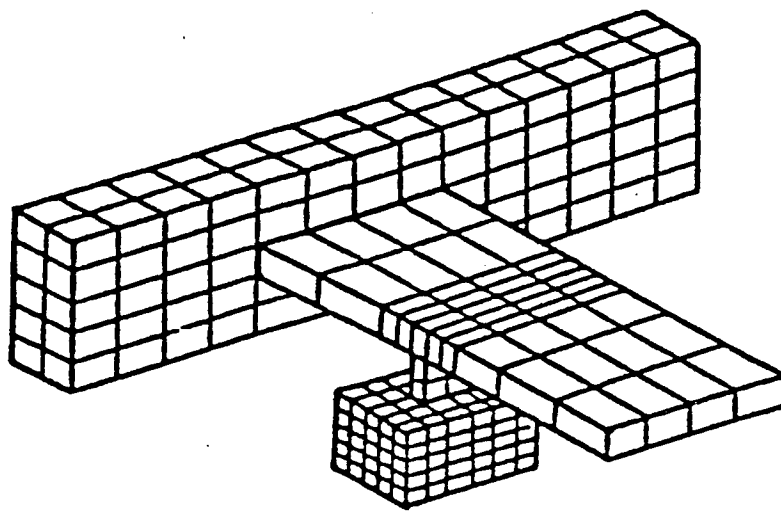
The main advantage of block-structured grids is their adaptability to complex configurations. However, special issues are introduced in the flow solution algorithms such as fictitious corners and nonanalytic block boundaries. These algorithm issues have been studied in two-dimensions (reference 3) and three-dimensions (reference 2) when the flow is solved by a finite volume method using a SLOR scheme.

Two innovative approaches for solving the transonic flow in a multi-block grid were explored. The first approach examines a method involving "zonal decomposition" wherein block boundaries are treated as true boundary surfaces separating interfacing grids. The issues investigated involve techniques for matching solutions at a block boundary. A feasibility study was completed and the results are presented.

The second approach (Figure 2) involves overlapping grids for differencing across a block boundary near an artificially induced coordinate singularity occurring at a fictitious corner. This approach selects a set of neighboring nodes for the fictitious corner (node 3, see Figure 2c) such that the resulting physical cells (cell 1342 and cell 3564) for node 3 are topologically the same as any other node on the airfoil surface.

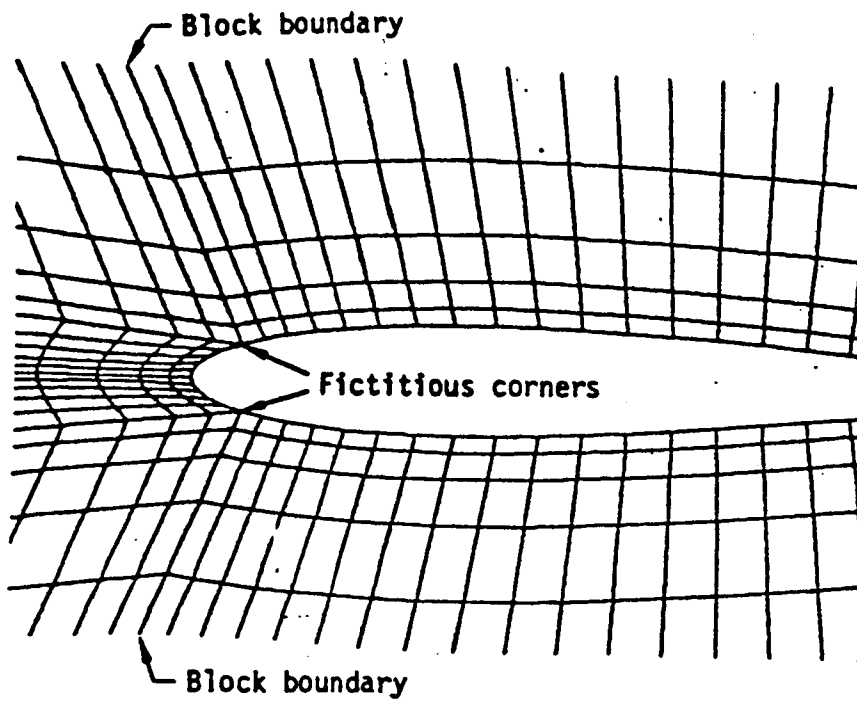


Physical space

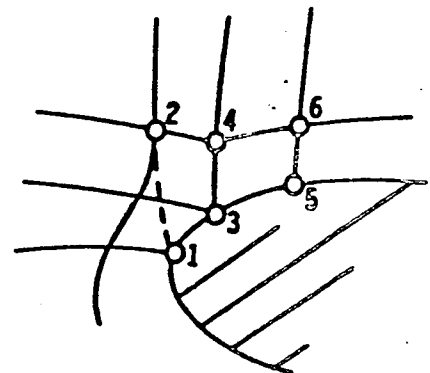


Computational space

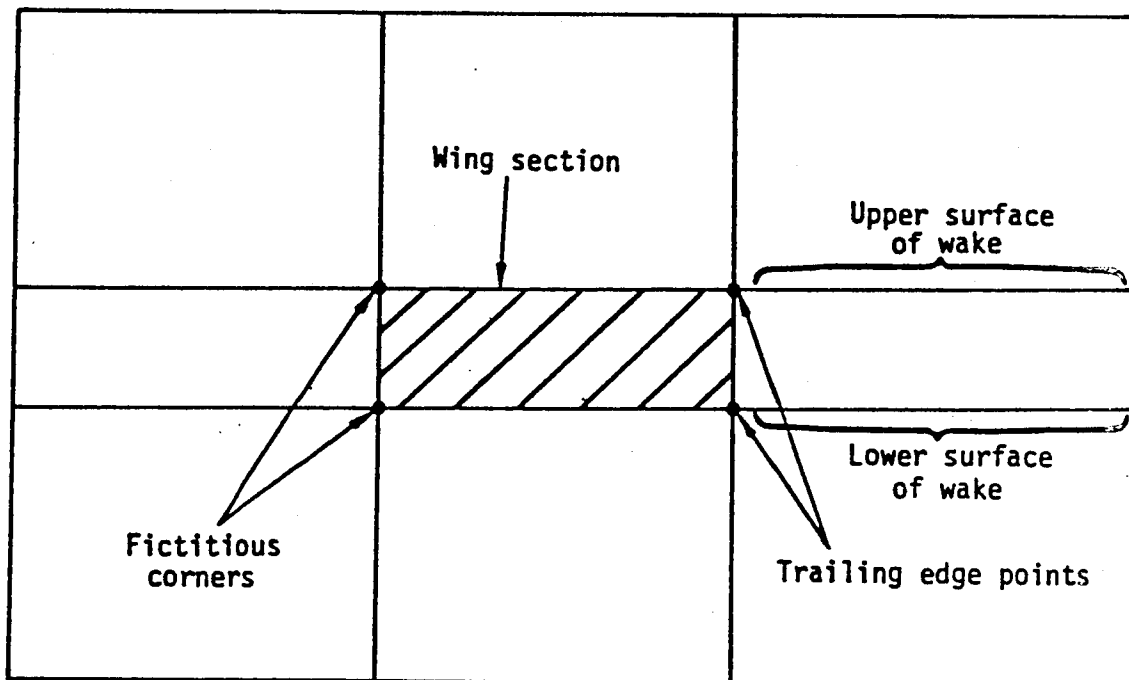
Figure 1. Block Structuring



(a) Physical space



(c) Near a fictitious corner



(b) Computational space

Figure 2. A Typical Block-Structured Grid.

To explore the second approach, an AF-based transonic code, called TWINGB, was modified for the treatment of flow using multi-block grids. During the course of this study, it was discovered that the flow solution algorithm in TWINGB is sensitive to grid effects. A method to improve the flow solution algorithm in TWINGB to reduce its grid dependency has been derived. The modified TWINGB has been evaluated on test cases with a typical parabolic arc airfoil and wing using block-structured grids. Results of the study indicate a further need to improve the flow solution algorithm.

The block-structured grid is discussed in Chapter II. The fully implicit AF scheme is given in Chapter III. The method of solution employing zonal decomposition is presented in Chapter IV. The study of grid effects for the AF-based TWINGB code is summarized in Chapter V. Results for adaptation of TWINGB to block-structured grids are presented in Chapter VI. Conclusions and recommendations are given in Chapters VII and VIII.

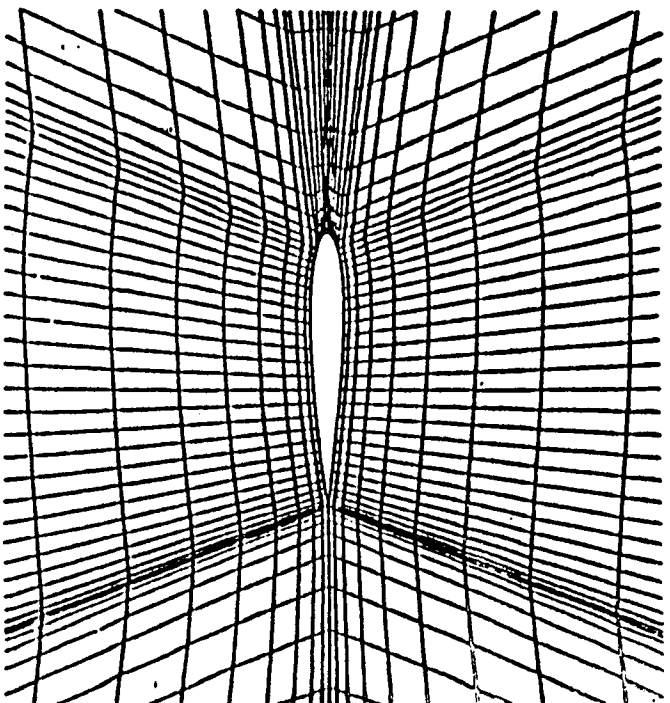
## II. BLOCK-STRUCTURED GRID

The construction of a suitable grid system for complex 3D configurations, such as a wing/body/nacelle, is a necessary step for computing the corresponding transonic full potential flow. Two approaches have been available based on Thompson's (reference 4) surface-adapted coordinate concept. A limited approach maps the flow domain surrounding a 3D configuration into a single rectangular box. This approach has been successful for simple geometries (reference 5) but cannot be effectively applied to complex configurations with multiple components. The more general approach divides the computational domain into multiple rectangular blocks where the configuration itself is also represented by a set of blocks whose structure follows the nature lines of the configuration (Figure 1). This results in a multi-block grid system which is adaptable to complex configurations and can produce good grid quality near physical corners. A comparison of typical single block and multi-block grids is given in Figure 3.

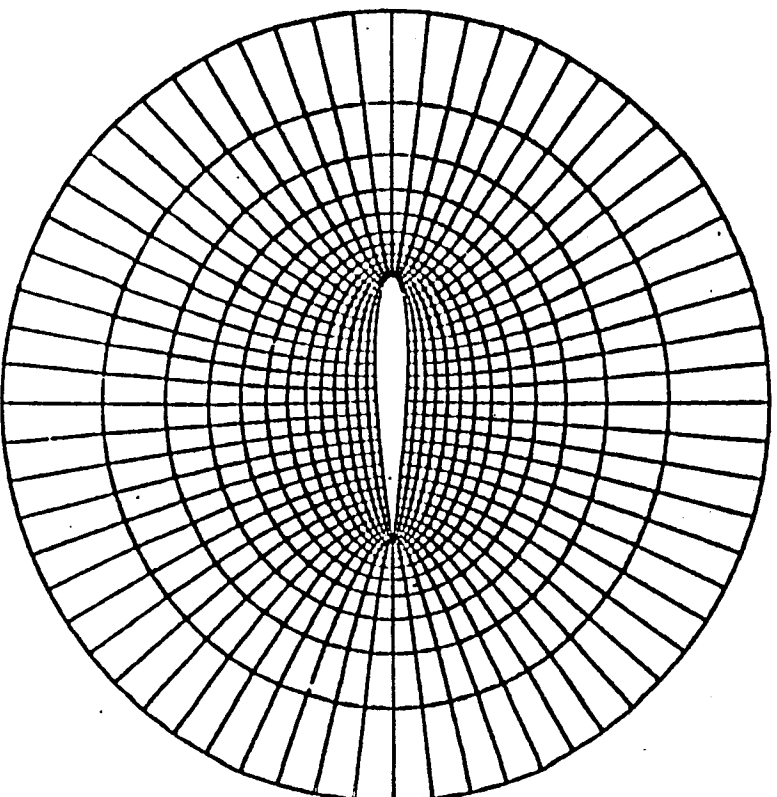
Generally, a single-block grid structure will not be suitable for geometrically complex boundary configurations. Thus, the flow field must be divided into multiple blocks to accommodate slope discontinuities of the boundary surfaces and provide good boundary fitting behavior (Figure 4). A typical block-structured grid generation (Figure 5) process is described below:

1. Define the overall block structure according to the natural lines of the configuration.
2. Generate 1D grid along the block perimeters (perimeter discretization).
3. Generate 2D grid covering the block surfaces using the information obtained in Step 2 as boundary data for the surface grid generation.
4. Generate 3D grid covering the volume grids filling each block using the information obtained in Step 3 as boundary data for the volume grid generation.

**MULTIPLE-BLOCK GRID**

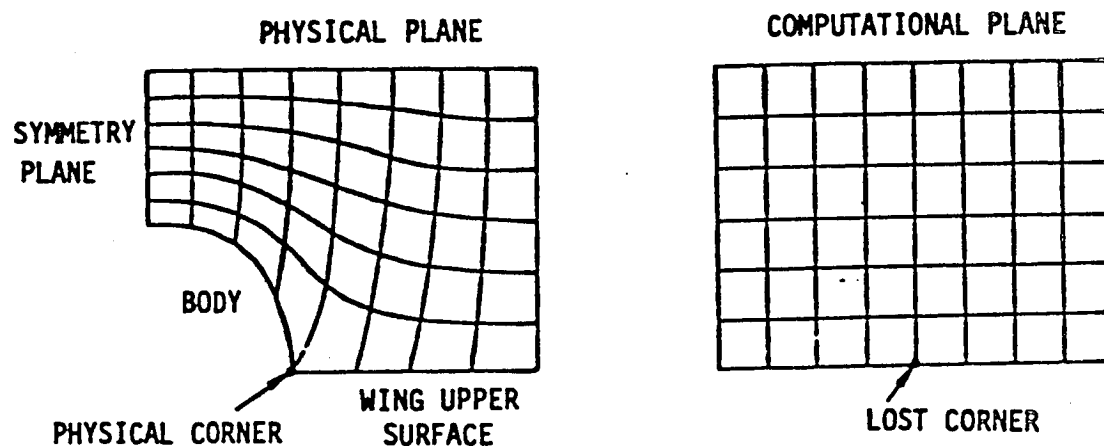


**SINGLE-BLOCK GRID**



**Figure 3. Comparison of Grids Near an Airfoil.**

(A) SINGLE-BLOCK GRID



(B) MULTIPLE-BLOCK GRID

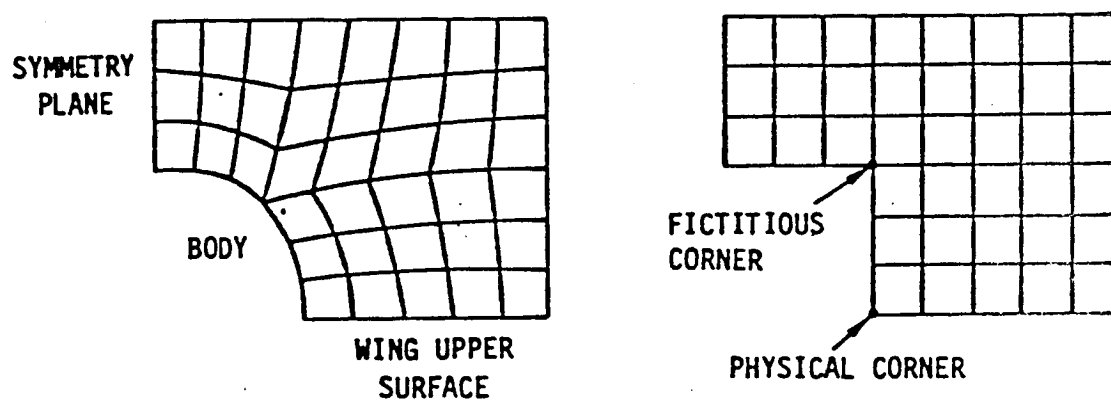


Figure 4. Comparison of Grid Structure.

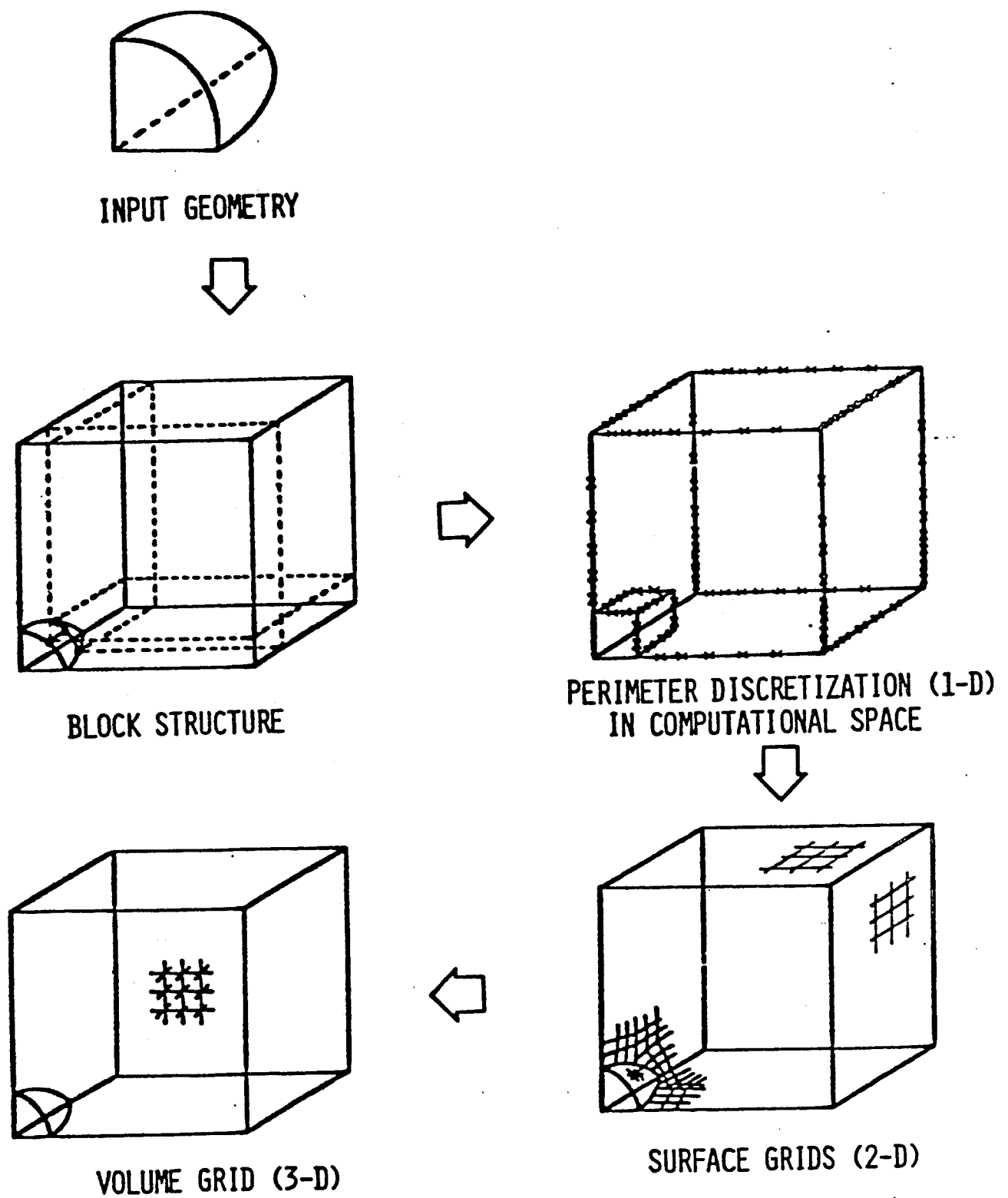


Figure 5. Block-Structured Grid Generation Process.



Thompson's method of generating a boundary-fitted curvilinear coordinate system requires that the curvilinear coordinates be solutions of a system of elliptic partial differential equations in the physical space subject to Dirichlet boundary conditions on all boundaries. One (curvilinear) coordinate is specified to be constant on each of the boundaries and a monotonic variation of the other coordinate along each boundary is specified. By interchanging the role of the Cartesian coordinates and the transformed coordinates, one arrives at a quasi-linear elliptic system for the Cartesian coordinates in the transformed space.

The present approach simplifies the formulation by requiring the Cartesian coordinates  $\bar{x} = (x, y, z)$  to be the solutions of three independent linear equations defined in the computational coordinates  $(\xi, \eta, \zeta)$ :

$$A\bar{x}_{\xi\xi} + B\bar{x}_{\eta\eta} + C\bar{x}_{\zeta\zeta} + D\bar{x}_{\xi} + E\bar{x}_{\eta} + F\bar{x}_{\zeta} = 0 \quad (1)$$

where the coefficients A to F are constants or specified functions used for grid control. The coefficients A, B, and C provide means to locally rescale the computational coordinates. Along the block boundaries, these three coefficients may be evaluated knowing the grid distributions, e.g.

$$A \propto (\bar{x}_{\eta} \times \bar{x}_{\zeta})^2 \quad (2)$$

The coefficients D, E, and F are related to the source terms in Thompson's nonlinear approach (reference 4). This can be seen by comparing equation (1) with the quasi-linear equations for 3D grid generations (more details can be seen in reference 2). These three coefficients can be extracted effectively from the perimeter discretization of the block boundary (reference 6) taking the limiting form of equation (1). For example, D may be extracted from the relation.

$$A\bar{x}_{\xi\xi} + D\bar{x}_{\xi} = 0 \quad (3)$$

The coefficients D, E, and F are linearly interpolated on a block boundary surface using the block perimeter value. The grid control coefficients A to F are then linearly interpolated inside a block using its boundary values.

### III. AF2 SCHEME

A fast, fully implicit AF algorithm to solve the conservative, transonic, full potential equation in 3D was presented by Holst (reference 1) in a single-block ring grid. General guidelines for the construction of implicit AF schemes follows from the two-level iteration procedure.

$$NC^n + \omega L\phi^n = 0 \quad (4)$$

where  $C^n = \phi^{n+1} - \phi^n$  is the correction,  $L\phi^n$  is the residual of the nth level velocity potential solution ( $\phi^n$ ), and  $\omega$  is a relaxation parameter.

In the AF approach, N is factorized by several factors, usually, two factors for 2D algorithms and three factors for 3D algorithms. These factors are chosen so that their product is an approximation to L, only simple matrix operations are required to ensure computational efficiency, and the overall scheme is stable.

Stability in the supersonic regions of flow has been achieved by adding an artificial viscosity term using an upwind bias of the density coefficient.

#### A. Governing Equations

The 3D full-potential equation in strong conservation-law form is given by

$$(\rho\phi_x)_x + (\rho\phi_y)_y + (\rho\phi_z)_z = 0 \quad (5a)$$

$$\rho = [1 - \frac{\gamma-1}{\gamma+1} (\phi_x^2 + \phi_y^2 + \phi_z^2)]^{\frac{1}{\gamma-1}} \quad (5b)$$

The density ( $\rho$ ) is normalized by the stagnation density ( $\rho_s$ ), the three velocity components ( $\phi_x$ ,  $\phi_y$ , and  $\phi_z$ ) are normalized by the critical sound speed; x, y, and z are Cartesian coordinates in the streamwise, spanwise,

and vertical directions respectively, and  $\gamma$  is the ratio of specific heats.

Equation (5) is transformed from the physical space  $(x, y, z)$  to the computational space  $(\xi, \eta, \zeta)$  using a general independent variable transformation. This transformation, denoted by

$$\xi = \xi(x, y, z) \quad \eta = \eta(x, y, z) \quad \zeta = \zeta(x, y, z) \quad (6)$$

maintains the strong conservation-law form of Equation (5). The full potential equation written in the computational domain is given by

$$(\rho U/D)_\xi + (\rho V/D)_\eta + (\rho W/D)_\zeta = 0 \quad (7a)$$

$$D = \nabla \xi \cdot (\nabla \eta \times \nabla \zeta) = \det(J) \quad (7b)$$

$$q^2 = \begin{Bmatrix} U \\ V \\ W \end{Bmatrix}^T \begin{Bmatrix} \phi_\xi \\ \phi_\eta \\ \phi_\zeta \end{Bmatrix} \quad (7c)$$

$$\rho = [1 - \frac{\gamma-1}{\gamma-1} q^2]^{\frac{1}{\gamma-1}} \quad (7d)$$

$$U = \nabla \xi \cdot \nabla \phi$$

$$V = \nabla \eta \cdot \nabla \phi \quad (8a)$$

$$W = \nabla \zeta \cdot \nabla \phi$$

$$\begin{Bmatrix} U \\ V \\ W \end{Bmatrix} = JJ^T \begin{Bmatrix} \phi_{\xi} \\ \phi_{\eta} \\ \phi_{\zeta} \end{Bmatrix} \quad (8b)$$

Let

$$JJ^T = \begin{Bmatrix} A_1 & A_4 & A_5 \\ A_4 & A_2 & A_6 \\ A_5 & A_6 & A_3 \end{Bmatrix} \quad (8c)$$

then

$$A_1 = \xi_x^2 + \xi_y^2 + \xi_z^2$$

$$A_2 = \eta_x^2 + \eta_y^2 + \eta_z^2$$

$$A_3 = \zeta_x^2 + \zeta_y^2 + \zeta_z^2$$

$$A_4 = \xi_x \eta_x + \xi_y \eta_y + \xi_z \eta_z \quad (11a)$$

$$A_5 = \xi_x \zeta_x + \xi_y \zeta_y + \xi_z \zeta_z$$

$$A_6 = \eta_x \zeta_x + \eta_y \zeta_y + \eta_z \zeta_z$$

and

$$\begin{Bmatrix} U \\ V \\ W \end{Bmatrix} = \begin{Bmatrix} A_1 & A_4 & A_5 \\ A_4 & A_2 & A_6 \\ A_5 & A_6 & A_3 \end{Bmatrix} \begin{Bmatrix} \phi_\xi \\ \phi_\eta \\ \phi_\zeta \end{Bmatrix} \quad (8e)$$

$U$  ( $V$  or  $W$ ) is the contravariant velocity component normal to the  $\xi$  ( $\eta$  or  $\zeta$ ) plane,  $A_1$  to  $A_6$  are matrix quantities, and  $J$  is the Jacobian matrix of the transformation. The expression in Equation (8) can be easily evaluated using the relation

$$J = \begin{Bmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{Bmatrix}^{-1} \quad (9)$$

Equation (9) is valid so long as the Jacobian matrix is nonsingular.

Equations (7) and (8) can be simplified if all constant  $\eta$  surfaces coincide with constant  $y$  planes. That is:

$$y = 0 \quad y_\eta = 0 \quad (10)$$

The metric quantities of Equations (8) and (9) become:

$$A_1 = \xi_x^2 + \xi_y^2 + \xi_z^2 \quad A_2 = \eta_y^2$$

$$\begin{aligned}
A_3 &= \xi_x^2 + \xi_y^2 + \xi_z^2 & A_4 &= \xi_y \eta_y \\
A_5 &= \xi_x \xi_x + \xi_y \xi_y + \xi_z \xi_z & A_6 &= \xi_y \eta_y \\
D &= 1/(x_\xi z_\xi - x_\xi z_\xi) \eta_n
\end{aligned} \tag{11a}$$

and

$$\begin{aligned}
\xi_x &= D(y_n z_\xi) & \xi_x &= -D(y_n z_\xi) \\
\xi_y &= D(x_\xi z_n - x_n z_\xi) & \xi_y &= D(x_n z_\xi - x_\xi z_n) \\
\xi_z &= -D(x_\xi y_n) & \xi_z &= D(x_\xi y_n) \\
\eta_x &= 0 \quad \eta_y = 1/y_n & \eta_z &= 0
\end{aligned} \tag{12}$$

## B. Flow Solution Algorithm

Let  $i, j, k$  be the subscripts indicating position in the  $\xi, \eta, \zeta$  computational mesh. Referring to a point  $i, j, k$ , a second order accurate finite difference approximation to equation (7a) is given by:

$$\begin{aligned}
&\delta_\xi^+ \left( \frac{\rho U}{D} \right)_{i+\frac{1}{2}, j, k} + \delta_\eta^+ \left( \frac{\rho V}{D} \right)_{i, j+\frac{1}{2}, k} \\
&+ \delta_\zeta^+ \left( \frac{\rho W}{D} \right)_{i, j, k+\frac{1}{2}} = 0
\end{aligned} \tag{13}$$

The operator  $\delta_\xi^+$  ( $\delta_\eta^+$  or  $\delta_\zeta^+$ ) is first order accurate backward difference operator in  $\xi$  ( $\eta$  or  $\zeta$ ) direction. These operators are defined by:

$$\begin{aligned}
\delta_\xi^+ ( )_{i, j, k} &= ( )_{i, j, k} - ( )_{i-1, j, k} \\
\delta_\eta^+ ( )_{i, j, k} &= ( )_{i, j, k} - ( )_{i, j-1, k} \\
\delta_\zeta^+ ( )_{i, j, k} &= ( )_{i, j, k} - ( )_{i, j, k-1}
\end{aligned} \tag{14}$$

It should be noted that equation (14) provides the divided difference formula since the standard divisor  $\Delta \xi$ ,  $\Delta \eta$ , and  $\Delta \zeta$  are all equal to one and have been eliminated. The density is calculated using equation (7d). Values of density are computed and stored at half-I point in the computational mesh. The  $\xi$ ,  $\eta$ , and  $\zeta$  derivatives of  $\phi$ , required for computing the density at the half-I point  $i+1/2, j, k$ , are approximated by:

$$\begin{aligned}
 \phi_{\xi i+1/2, j, k} &\doteq \phi_{i+1, j, k} - \phi_{i, j, k} \\
 \phi_{\eta i+1/2, j, k} &\doteq 0.25(\phi_{i+1, j, k} - \phi_{i+1, j-1, k} \\
 &\quad + \phi_{i, j+1, k} - \phi_{i, j-1, k}) \\
 \phi_{\zeta i+1/2, j, k} &\doteq 0.25(\phi_{i+1, j, k+1} - \phi_{i+1, j-1, k-1} \\
 &\quad + \phi_{i, j+1, k+1} - \phi_{i, j-1, k-1})
 \end{aligned} \tag{15}$$

The contravariant velocity components,  $U_{i+1/2, j, k}$ ,  $V_{i, j+1/2, k}$ , and  $W_{i, j, k+1/2}$  are computed by standard second order accurate finite difference formulas, e.g.,

$$\begin{aligned}
 U_{i+1/2, j, k} &= A_{1 i+1/2, j, k} \phi_{\xi i+1/2, j, k} \\
 &\quad + A_{4 i+1/2, j, k} \phi_{\eta i+1/2, j, k} \\
 &\quad + A_{5 i+1/2, j, k} \phi_{\zeta i+1/2, j, k}
 \end{aligned} \tag{16}$$

where the approximation formulas in equation (15) are used to provide the difference approximations in  $\xi$ ,  $\eta$  and  $\zeta$  for  $\phi$ .

In order to maintain stable supersonic region, the following artificial viscosity term was added to equation (13)

$$-\Delta\xi(v\rho \frac{|U|}{D})_{\xi} - \Delta\eta(v\rho \frac{|V|}{D})_{\eta} - \Delta\zeta(v\rho \frac{|W|}{D})_{\zeta} \quad (17)$$

where  $v$  is an artificial viscosity coefficient to be defined later. The addition of such an artificial viscosity term is equivalent to retarding the density in the original centrally differenced scheme equation (13). This leads to

$$\delta_{\xi}(\frac{\tilde{\rho}U}{D})_{i+\frac{1}{2},j,k} + \delta_{\eta}(\frac{\tilde{\rho}V}{D})_{i,j+\frac{1}{2},k} + \delta_{\zeta}(\frac{\tilde{\rho}W}{D})_{i,j,k+\frac{1}{2}} = 0 \quad (18a)$$

where:

$$\tilde{\rho}_{i+\frac{1}{2},j,k} = [(1-v)\rho]_{i+\frac{1}{2},j,k} + v_{i+\frac{1}{2},j,k} \rho_{i+r+\frac{1}{2},j,k} \quad (18b)$$

$$\tilde{\rho}_{i,j+\frac{1}{2},k} = [(1-v)\rho]_{i,j+\frac{1}{2},k} + v_{i,j+\frac{1}{2},k} \rho_{i,j+s+\frac{1}{2},k} \quad (18c)$$

$$\tilde{\rho}_{i,j,k+\frac{1}{2}} = [(1-v)\rho]_{i,j,k+\frac{1}{2}} + v_{i,j,k+\frac{1}{2}} \rho_{i,j,k+t+\frac{1}{2}} \quad (18d)$$

The  $r$ ,  $s$ , and  $t$  indices are selected to ensure the  $\xi$ ,  $\eta$ , and  $\zeta$  differencing of  $\rho$  in equation (17) are evaluated using upwind formulas. Thus,

$$\begin{aligned} r &= \begin{cases} -1 & \text{when } U_{i+1/2,j,k} > 0 \\ 1 & \text{when } U_{i+1/2,j,k} \leq 0 \end{cases} \\ s &= \begin{cases} -1 & \text{when } V_{i+1/2,j,k} > 0 \\ 1 & \text{when } V_{i+1/2,j,k} \leq 0 \end{cases} \\ t &= \begin{cases} -1 & \text{when } W_{i+1/2,j,k} > 0 \\ 1 & \text{when } W_{i+1/2,j,k} \leq 0 \end{cases} \end{aligned} \quad (19)$$



This maintains an upwind influence in the differencing scheme for supersonic regions for arbitrary mesh and arbitrary velocity vector. The artificial viscosity coefficient  $\nu$  is computed by:

$$\nu_{i+\frac{1}{2},j,k} = \begin{cases} \text{MAX}[(M_{i,j,k}^2 - 1)C, 0] & \text{for } U_{i+\frac{1}{2},j,k} > 0 \\ \text{MAX}[M_{i+\frac{1}{2},j,k}^2 - 1)C, 0] & \text{for } U_{i+\frac{1}{2},j,k} < 0 \end{cases} \quad (20)$$

The parameter  $C$  is a constant specified by the users.  $C$  is usually set between 1.0 and 2.0. Use of larger  $C$  increases the artificial viscosity (upwinding) in the difference scheme. An additional constraint is  $\nu \leq 1$  to improve the stability (reference 1). Expressions for  $\nu$  at  $i,j+1/2,k$  and  $i,j,k+1/2$  can be written similar to equation (20).

### C. AF2 Algorithm

The AF2 fully implicit approximation factorization scheme (reference 1) for three-dimension full-potential equation can be expressed by choosing the  $N$  operator of equation (2) to be

$$\begin{aligned} \alpha N_{i,j,k}^n = & - \left[ \left( \alpha - \frac{1}{A_k} \delta_n^+ A_j \delta_n^+ \right) \left( A_k - \frac{1}{\alpha \delta_s} A_i \delta_s^+ \right) \right. \\ & \left. - \alpha E_s^{+1} A_k \right] (\alpha + \delta_s^+) C_{i,j,k}^n \end{aligned} \quad (21a)$$

where

$$A_i = (\hat{\rho} A_1 / D)_{i-\frac{1}{2},j,k}^n$$

$$A_j = (\bar{\rho} A_2 / D)_{i,j-\frac{1}{2},k}^n \quad (21b)$$

$$A_k = (\hat{\rho} A_3 / D)_{i,j,k-\frac{1}{2}}^n \quad (21c)$$

The  $\alpha$  is a free parameter defined subsequently, and the operator  $E_{\xi}^{+1}$  is a shifting operator in  $\xi$  given by:

$$E_{\xi}^{+1}(\quad)_{i,j,k} = (\quad)_{i,j,k+1} \quad (22)$$

The AF2 scheme can be implemented in a three-step form given by:

### Step 1

$$(\alpha - \frac{1}{A_k} \vec{\delta}_n A_j \vec{\delta}_n) g_{i,j}^n = \alpha \omega L \phi_{i,j,k}^n + \alpha A_{k+1} f_{i,j,k+1}^n \quad (23)$$

### Step 2

$$(A_k - \frac{1}{\alpha} \vec{\delta}_{\xi} A_i \vec{\delta}_{\xi}) f_{i,j,k}^n = g_{i,j}^n \quad (24)$$

### Step 3

$$(\alpha + \vec{\delta}_{\xi}) C_{i,j,k}^n = f_{i,j,k}^n \quad (25)$$

Here  $\omega$  is a relaxation parameter to be defined by the user with a default value of 1.8. Steps 1 and 2 are solved for each  $k$  plane in the decreasing  $k$  direction. This provides  $f_{i,j,k}$  for all grid points. Step 3 is then solved in the increasing  $k$  direction for  $C_{i,j,k}$ .

Within each factorized step, the boundary conditions for the intermediate solutions (e.g.,  $g$  and  $f$ ) should be specified. Initiation of Step 1 at  $S = S_{\max}$  ( $k = NK$ ) plane requires information of  $f$  at  $NK+1$  which is unobtainable. A solution is to set  $f$  at  $NK+1$  equal to zero. Similarly, a boundary condition is implemented for  $g$  at  $n = n_{\min}$  ( $j = 1$ ) and  $n = n_{\max}$  ( $j = NJ$ ) by imposing  $(g_n)_{i,1} = (g_n)_{i,NJ} = 0$ .

For block-structured grid, a boundary condition for  $f$  is required at  $\xi = \xi_{\min}$  ( $i = 1$ ) and  $\xi = \xi_{\max}$  ( $i = NI$ ). This can be implemented by

imposing  $f_{1,j,k} = f_{NI,j,k} = 0$ . This is consistent with the AF2 scheme written in correction form, since convergence implies zero corrections everywhere and, therefore, zero  $f$  everywhere.

#### D. Temporary Damping

For the  $N$  operator in the AF2 scheme, one of the  $\xi$ ,  $\eta$ , or  $S$  difference approximation must be split between two factors. This generates either a  $\phi_{\xi t}$ ,  $\phi_{\eta t}$ , or  $\phi_{St}$  type term, and if it is properly upwind differenced (reference 7), provides time-dependent dissipation to the convergence process. Following from reference 8, the  $\xi$  difference approximation is split between two factors. The control over  $\eta$  (or  $\xi$ ) coordinate direction is achieved by adding  $\phi_{\eta t}$  (or  $\phi_{\xi t}$ ) term explicitly in Step 1 (or Step 2). Further detail can be found in reference 1.

The  $\alpha$  in equation (21a) can be considered as  $1/\Delta t$ . This direct analogy to time suggests the use of large time step (i.e., small  $\alpha$ ) to advance time as fast as possible for obtaining fast convergence. Reference 9 pointed out that this is effective for attacking the low frequency errors but not the high frequency errors. An effective approach is to use an  $\alpha$  sequence containing several values of  $\alpha$ . The small values are particularly effective for reducing the low frequency errors, and the large values are particularly effective for reducing the high frequency errors. A suitable  $\alpha$  sequence with analytical estimated end points ( $\alpha_L$ ,  $\alpha_H$ ) is given in reference 9. The sequence is generated using the end points to form a geometric sequence.

#### E. Boundary Condition

Along a solid surface, flow tangency boundary condition is imposed. This implies that the relevant contravariant velocity component should be zero which is simulated by a reflection procedure to account for the flux balance. The zeroing of this component also provides a relation to evaluate the derivative of  $\phi$  along a grid line normal to the solid surface. This eliminates the need to use one-sided difference on  $\phi$ . At the outer boundary of the computational mesh, ( $\xi = \xi_{\min}$ ,  $\xi = \xi_{\max}$ , or  $S = S_{\min}$ ) the  $\phi$  and  $\rho$  are assigned at the freestream values.



#### IV. SOLUTION BY ZONAL DECOMPOSITION

Block-structured grids are effective for discretizing the computational domain. Within each block a grid of good quality can be generated. The overall block structuring generally leads to nonanalytical block boundaries and nonstandard grid cells. An interesting way to treat this class of grids is to solve the flow by zonal decomposition using interfacing grids. The problems with the nonanalytical grid block boundaries and nonstandard grid cells will be removed with the implementation of suitable inter-block continuity conditions between neighboring blocks of grid in the flow solution algorithm. In addition, noncontiguous interfacing grids may be used to increase the flexibility of the grids.

##### A. Model Problem

The test problem formulated to explore the issues of matching is shown in the first part of Figure 6. A rectangular domain AEBCFD was chosen, with the 2D Laplace's equation:

$$L\phi = \phi_{xx} + \phi_{yy} = 0 \quad (26)$$

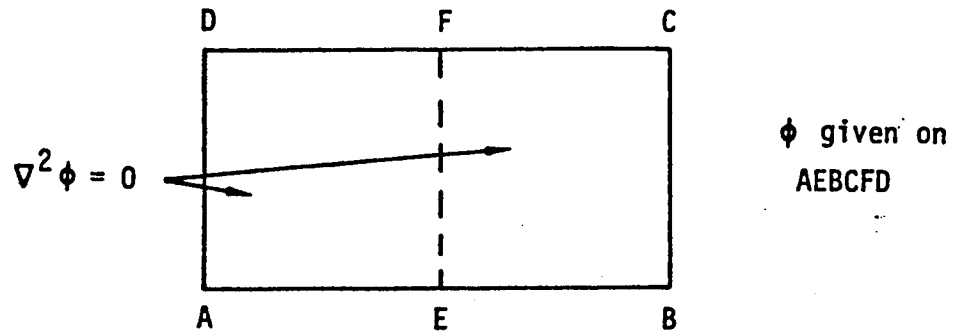
defined on a rectangular box

$$0 \leq x \leq 2, \quad 0 \leq y \leq 1$$

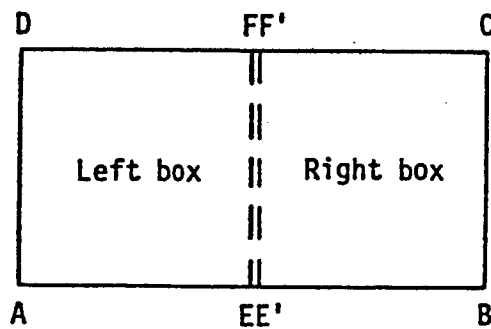
On the box boundary, boundary conditions were chosen of the form

$$\phi = \sinh(x)\sin(y) + \cosh(x)\cos(y) \quad (27)$$

A boundary EF dividing the box into two domains was defined. Solutions for left and right sides were computed independently and matched across EF by the procedure discussed below. These solutions were then compared with the exact solution calculated for a single domain comprising the large box with the boundary EF deleted.



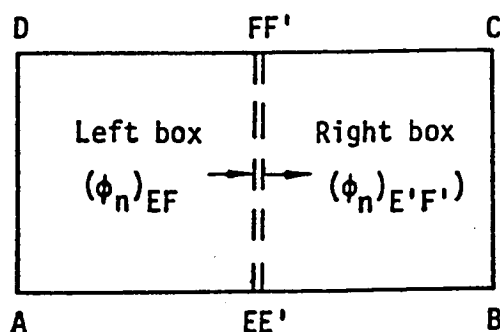
Step 1 To start



An initial estimation of  $\phi$  is assigned along EF and  $E'F'$  where  $\phi_{EF} = \phi_{E'F'}$ . Solve left box and right box separately. Compute  $(\phi_n)_{EF}$  from left box,  $(\phi_n)_{E'F'}$  from right box. Define  $\bar{\phi}_n = \frac{1}{2}[(\phi_n)_{EF} + (\phi_n)_{E'F'}]$ .

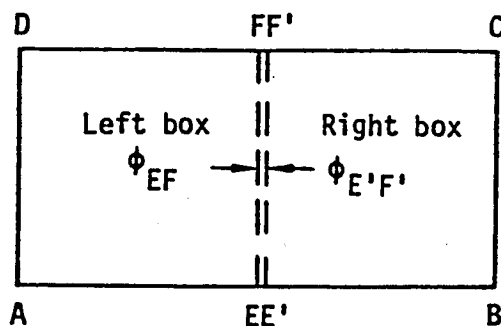
Figure 6. Alternating Boundary Conditions Flow Field Matching.

## Step 2 Neumann Boundary Condition Matching



Neumann Boundary Condition is imposed along EF,  $(\phi_n)_{EF} = \bar{\phi}_n$ , and along E'F',  $(\phi_n)_{E'F'} = \bar{\phi}_n$ . Again we solve left and right boxes separately and define  $\bar{\phi} = \frac{1}{2}(\phi_{EF} + \phi_{E'F'})$ .

## Step 3 Dirichlet Boundary Condition Matching



Dirichlet Boundary Condition is imposed along EF and E'F',  $\phi_{EF} = \bar{\phi} = \phi_{E'F'}$ , and left and right boxes are solved separately. Define  $\bar{\phi}_n = \frac{1}{2}[(\phi_n)_{EF} + (\phi_n)_{E'F'}]$ , go back to Step 2 until solution converges.

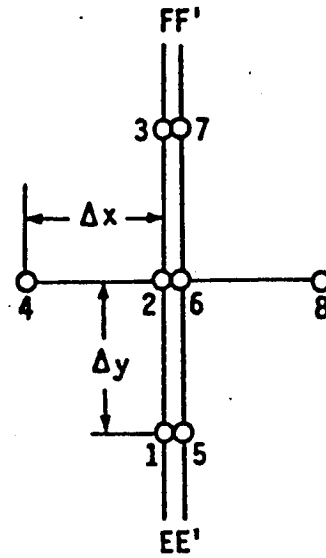
Figure 6. (Continued)

## B. Flow Field Matching Procedure

The flow field is divided into two boxes as illustrated in Figure 6. The flow in each box is solved by an AF scheme. The matching scheme used involved a sequential application of alternating boundary conditions (termed an ABC method) as outlined in the sequential part of Figure 6.

## C. Calculation of $\phi_n$ Along EF

In the first example, grids which were contiguous at EF were used. For either the left or right box, along the matching surface, the normal derivative  $\phi_x$  need be computed using information coming from one side of the surface. We note that the normal to the matching surface is aligned with x. For the left box,  $\phi_x$  is computed at node 2 (see adjacent sketch). We construct a difference approximation to equation (26) and require that this equation be satisfied at node 2. This leads to



$$\frac{1}{\Delta y^2} (\phi_3 - 2\phi_2 + \phi_1) + \frac{1}{0.5\Delta x} [(\phi_x)_2 - \frac{\phi_2 - \phi_4}{\Delta x}] = 0 \quad (28)$$

solving for  $(\phi_x)_2$  gives

$$(\phi_x)_2 = \frac{\phi_2 - \phi_4}{\Delta x} - \frac{0.5\Delta x}{\Delta y^2} (\phi_3 - 2\phi_2 + \phi_1) \quad (29)$$



For the right box at point 6, a similar formula can be derived

$$(\phi_x)_6 = \frac{\phi_8 - \phi_6}{\Delta x} + \frac{0.5\Delta x}{\Delta y^2} (\phi_7 - 2\phi_6 + \phi_5) \quad (30)$$

When convergence between right and left solutions is achieved, both the solutions and its normal derivative should be continuous across the matching surface.

$$\phi_1 = \phi_5, \quad \phi_2 = \phi_6, \quad \phi_3 = \phi_7 \quad (31)$$

$$\text{and} \quad (\phi_x)_2 = (\phi_x)_6$$

This implies that

$$\frac{1}{\Delta x^2} (\phi_8 - 2\phi_2 + \phi_4) + \frac{1}{\Delta y^2} (\phi_3 - 2\phi_2 + \phi_1) = 0$$

This is consistent with the requirement that the finite difference equation for Laplace's equation be satisfied at node 2.

#### D. AF Scheme

An iterative scheme for Laplace's equation can be written in correction form

$$NC^n + \omega R^n = 0$$

$$C^n = \phi^{n+1} - \phi^n \quad (32)$$

$$R^n = L\phi^n = (\delta_{xx} + \delta_{yy})\phi^n$$

where  $\delta_{xx}\phi$  and  $\delta_{yy}\phi$  are the central difference approximations for  $\phi_{xx}$  and  $\phi_{yy}$  respectively,  $L$  is the difference approximation of Laplace's operator,  $\omega$  is a relaxation factor. The matrix  $N$  is factorized such that

$$\alpha n C_{i,j}^n = - (\alpha - \delta_{yy})(\alpha - \delta_{xx}) C_{i,j}^n = - \alpha \omega L \phi^n \quad (33)$$

The flow field is solved in two steps.

#### Step 1

$$(\alpha - \delta_{yy}) f_{i,j}^n = \alpha \omega L \phi_{i,j}^n = \alpha \omega R^n \quad (34)$$

The intermediate variable  $f_{i,j}$  is solved from equation (34).

#### Step 2

$$(\alpha - \delta_{xx}) C_{i,j}^n = f_{i,j}^n \quad (35)$$

The correction vector  $C_{i,j}$  is solved from equation (35). The  $\omega$  is set equal to 2. The following formula (reference 10) is used to compute a sequence of  $N$   $\alpha$ .

$$\begin{aligned} \alpha_{\min} &= 1 \\ \alpha_{\max} &= \frac{4}{\Delta y^2} \\ \alpha_n &= \left( \frac{\alpha_{\min}}{\alpha_{\max}} \right)^{\frac{n-1}{N-1}} \alpha_{\max} \end{aligned} \quad (36)$$

#### E. Baseline Solutions

For the boundary condition given by equation 27, the exact solution for Laplace's equation is

$$\phi_{\text{ex}} = \sinh(x) \sin(y) + \cosh(x) \cos(y) \quad (37)$$

Numerical solutions for the same problem, on the entire flow field without using matching, were also computed on a coarse uniform grid consisting of 10 x 5 cells and on a fine uniform grid consisting of 20 x 10 cells.

For the coarse grid, after 15 iterations, the maximum residual reduced by seven orders of magnitude and the maximum error in  $\phi$  (compared with  $\phi_{ex}$ ) was 0.00174. For the fine grid, it took 16 iterations to achieve the same reduction of the maximum residual. The maximum error in  $\phi$  was 0.00045 for that case.

The numerical solution is highly converged and the errors in  $\phi$  reflect the truncation error of the finite difference approximation. These numerical solutions without using the flow field matching procedure will be referred to as baseline solutions.

#### F. Flow Field Matching With Contiguous Interfacing Grids

This simple example is effective for illustrating the approach of flow field matching. Refer to Figure 6, a coarse grid with 5 x 5 cells for the left box and a coarse grid with 5 x 5 cells for the right box was used. One AF sweep was used each time the boundary condition at the matching surface was changed. Thus, due to the alternation of boundary conditions, each iteration for flow field matching consists of two AF sweeps. It took 24 iterations for the maximum residual to reduce by seven orders of magnitude. The maximum error in  $\phi$  was 0.00174, which is the same as that in the baseline solution for the coarse grid. The same matching problem was also computed on a fine grid in which the left and right boxes each had 10 x 10 cells. It took 39 iterations to reduce the maximum residual by seven orders of magnitude. The maximum error in  $\phi$  was 0.00045, the same as that in the baseline solution for the fine grid. Comparisons of convergence histories are given in Figure 7 for the coarse mesh and in Figure 8 for the fine mesh. Each iteration in the matching solutions take twice as many operations as the baseline solutions.

Thus the basic matching ideas were demonstrated. Fully converged results of accuracy equal to that of a single block situation were obtained, but the number of iterations required was significantly greater.

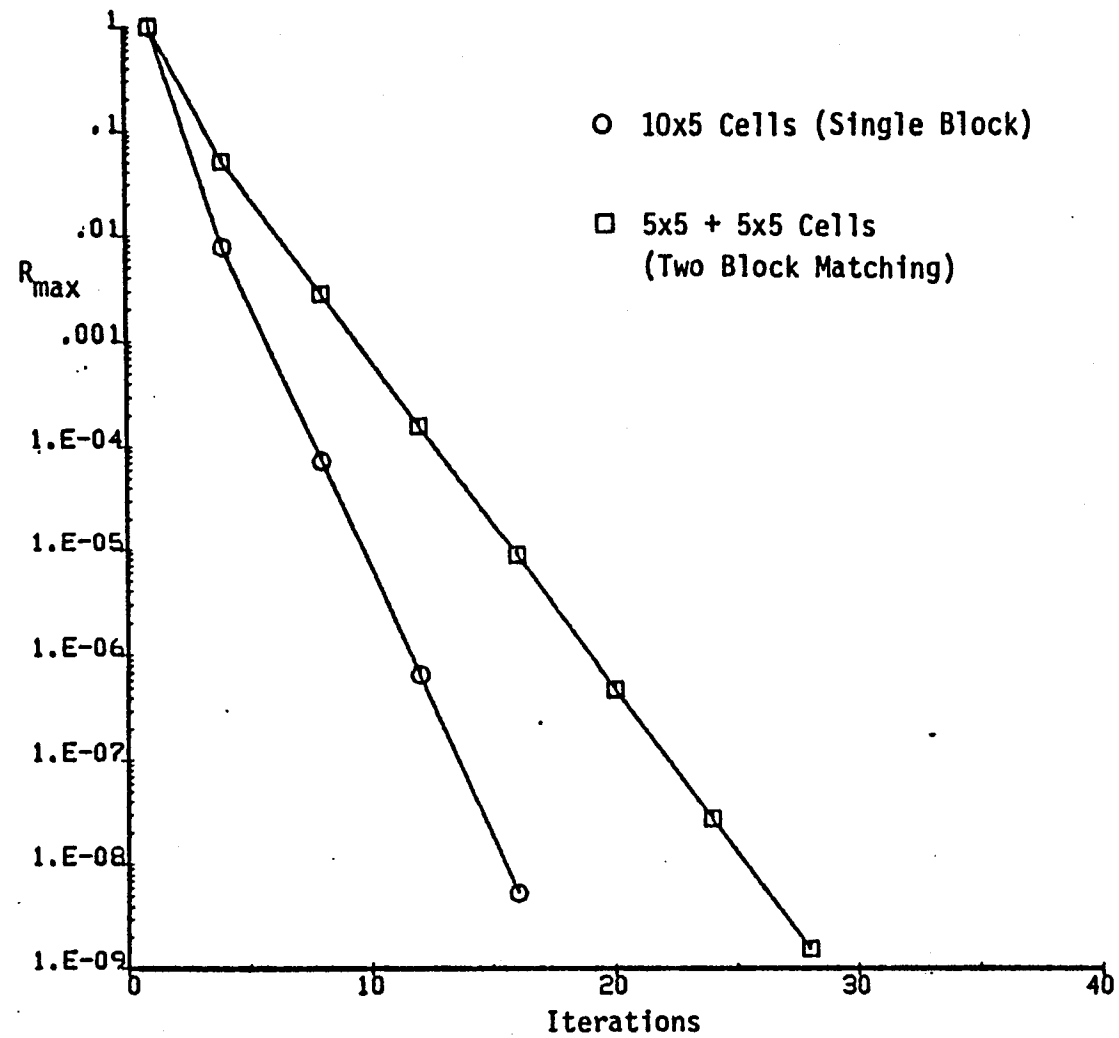


Figure 7. Convergence Histories on Coarse Mesh.

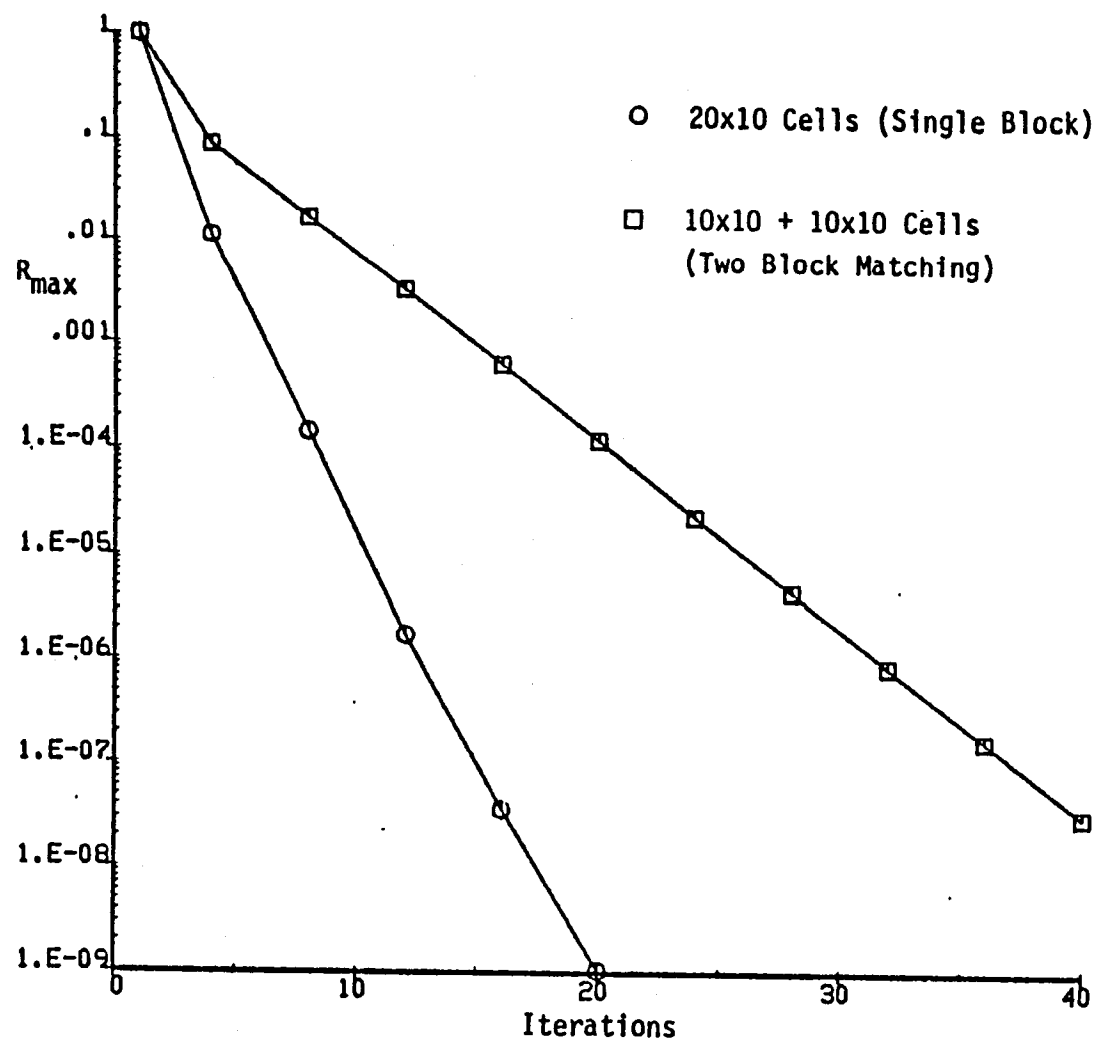


Figure 8. Convergence Histories on Fine Mesh.

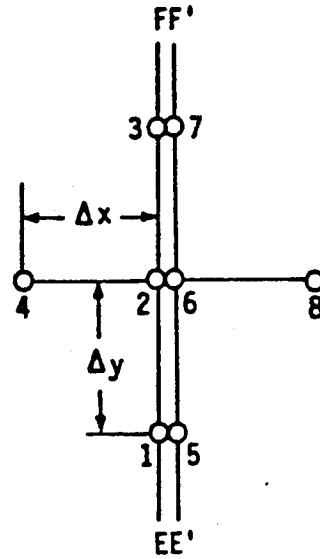
## G. Flow Field Matching With Noncontiguous Interfacing Grids

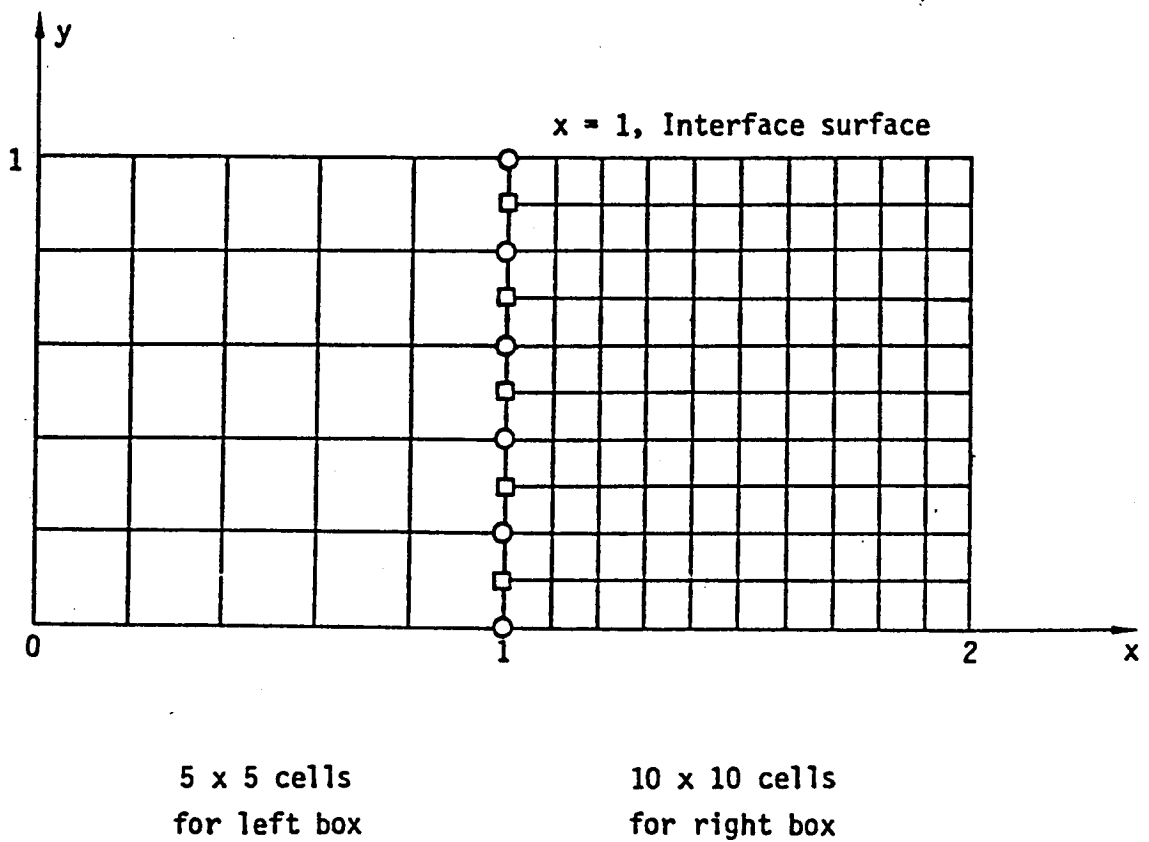
To further explore the versatility of the flow field matching approach, we studied a more difficult case where the left box and the right box have different grid resolutions. The left box consisted of  $5 \times 5$  cells and the right box had  $10 \times 10$  cells (Figure 9). Consequently, the two boxes are matched with noncontiguous interfacing grids. This test case is significant in that effective application of noncontiguous interfacing grids can reduce the number of grid points in the farfield and concurrently reduce the high aspect ratio grid in those regions. Grid quality will be improved and total number of unknown can be reduced to achieve good numerical accuracy and high computational efficiency. For the noncontiguous interfacing grids case, special treatment of the flow solution algorithm is needed at the matching surface. Refer to Figure 9, along the matching surface there are two types of nodes. A node which is common to both the left box and the right box is called a common node. A node belonging to the right box only is called an uncommon node.

For a common node, we first discuss the computation of  $\phi_x$  from the solution of the Dirichlet boundary condition part of the matching problem for point 2 (see adjacent sketch). The  $\phi_x$  is computed using equation 29, where  $\Delta x$  and  $\Delta y$  are replaced by  $\Delta x_1$  and  $\Delta y_1$  respectively. For point 6,  $\phi_x$  is computed using equation (30) with  $\Delta x$  and  $\Delta y$  replaced by  $\Delta x_2$  and  $\Delta y_2$ . Here

$$\Delta x_2 = 0.5\Delta x_1 \quad \text{and} \quad \Delta y_2 = 0.5\Delta y_1 \quad (38)$$

The average quantity  $\bar{\phi}_x$  used in the iteration procedure outline in Figure 6 is then obtained through averaging the corresponding values at node 2 and node 6.



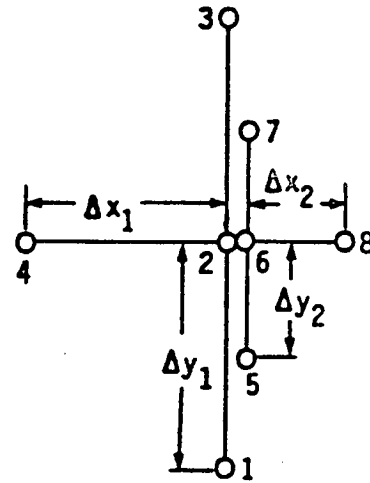


- Common node: A node which is common to both the left box and the right box.
- Uncommon node: A node which belongs to the right box only.

Figure 9. Flow Field Matching With Noncontiguous Interfacing Grids.

The solution of the Neumann boundary condition part of the cycle provides  $\phi_2$  and  $\phi_6$ , and  $\bar{\phi}$  is computed through averaging of  $\phi_2$  and  $\phi_6$ . This defines the boundary condition for the Dirichlet boundary condition part of the matching problem.

We next consider the uncommon node assuming that the solutions of the Dirichlet boundary condition problems are given (see adjacent sketch). For node 6,  $\phi_x$  is computed using equation (30) while for point 2,  $\phi_x$  was computed by averaging the  $\phi_x$  at node 1 and node 3. The  $\bar{\phi}_x$  can then be defined as the average of  $(\phi_x)_2$  and  $(\phi_x)_6$ .



When  $\phi$  is solved in the Neumann boundary condition part of the cycle,  $\phi$  is directly available at node 6 but not at point 2. We may define  $\phi_2$  by averaging  $\phi_1$  and  $\phi_3$ . The  $\bar{\phi}$  is then defined by averaging  $\phi_2$  and  $\phi_6$ .

Numerical experiments show that this scheme encounters a limit cycle. The maximum residual can only be reduced by less than two order of magnitude and then it starts to oscillate periodically. The oscillation is periodic since the sequence of acceleration parameters is changed in a periodic manner. The point at which the residual attains its maximum is always at an uncommon node. The maximum error in  $\phi$  can only be reduced to the order of 0.01 and this error also oscillates periodically. This error also reaches its maximum at an uncommon node.

In order to analyze why the solutions run into a limit cycle, we take a further study at the uncommon node. Refer to node 6, the residual can be reduced to zero at node 6 in the right box but not at point 2 in the left box. If we force  $\phi$  and  $\phi_x$  to be matched between point 2 and node 6, then there is no guarantee that the residual at node 6 can be reduced to zero in the flow field matching procedures. In essence, there are not enough degrees of freedom in the system to require both  $\phi$  and  $\phi_x$  to match exactly at uncommon nodes.



To correct this problem for the uncommon nodes, we can impose a continuity condition for either  $\phi$  or  $\phi_x$  but not both. Numerical experimentation was subsequently conducted by matching  $\phi_x$  at the uncommon nodes. Results indicate that this correction resolves the limit cycle problem. In addition, we used a weighted average procedure for the  $\phi$  (and  $\phi_x$ ) across the matching surface to replace the average procedure. This improved the convergence rate.

With this correction, the maximum residual reduced by seven orders of magnitude in 34 iterations. The residual still reaches its maximum at an uncommon node. The maximum error in  $\phi$  is  $-0.00190$  at an uncommon node. This is to be expected since at an uncommon node we only enforce the continuity of  $\phi_x$  and we allow  $\phi$  to have a jump across the matching surface.

Alternatively, one can impose continuity of  $\phi$  at the uncommon nodes and allow  $\phi_x$  to be discontinuous.

#### H. Conclusions on Zonal Decomposition

A summary of the zonal decomposition case study is given in Table 1.

Table 1 Zonal Decomposition Case Study Summary

Grid	Interfacing Grids Contiguous	Convergence in $R_{\max}$ to $10^{-7}$	Error
One block 10x5		15	0.00174
Two block 5x5+5x5	Yes	24	0.00174
One block 20x10		16	0.00045
Two block 10x10+10x10	Yes	39	0.00045
Two block 5x5+10x10	No	34	0.00190

Our overall conclusion at this point is that the method of matching outlined above is feasible. If grids of equal spacing interface at the matching boundary, the technique of matching does not cause a loss in accuracy. It does require more computer time than a single global solution.

We have also shown a capability for matching using noncontiguous interfacing grids (Figure 9). Our conclusion here is that the local accuracy at the matching boundary is of the order associated with the coarser of the interfacing grids, which is to be expected.

## I. Remaining Issues

There remain several issues to explore. One would entail a study of the interactions involved when a large number of computational domains interact through several matching boundaries. A second issue arises when one uses a compressible flow equation. In this case, the suitable matching conditions are the continuity of  $\phi$  and the normal mass flux. A third issue arises when one considers the use of a governing equation capable of displaying mixed character such as the transonic equation. In this case, the appropriate boundary conditions governing the solution in a single domain are not known a priori, as is the case with the simple Neumann or Dirichlet choices associated with Laplace's equation. In the transonic case, the options are: (i) two boundary conditions required (when supersonic flow is entering through a surface with a direction such that the local Mach cone enters the domain), (ii) one boundary condition required (either the local flow is subsonic, or in a supersonic case, the local Mach cone cuts the adjacent boundary surface) which can be either Neumann or Dirichlet, or (iii) no boundary condition allowed (as when the flow is leaving the domain at supersonic speed with no portion of the local Mach cone cutting the adjacent boundary surface). Since just a prescription of boundary values about a domain does not, in general, reveal whether the local direction or Mach number is such that two, one or no boundary conditions are allowed, one must begin to think in terms of adaptive or selective boundary conditions to be selected during the solution process according to the evolution of the local flow.

## V. GRID EFFECT STUDY ON TWINGB CODE

Computational experiences of TWINGB reported in reference 1 are based on a single-block ring grid. A typical ring grid can be seen in Figure 3. Such a grid is very effective for simple geometries, such as a simplified wing. The resulting grids for such simple problems are surface adapted, smoothly varying, nearly orthogonal, with good cluster control, containing no singularities, and the grid cell aspect ratio is moderate everywhere.

Some of these characteristics may not carry over when we use the block-structured grid. Such grid structures can produce fictitious corners, non-smooth block boundaries, and grid cells of high aspect ratio. The algorithm issues in the flow solution scheme relative to this class of grids have not been addressed previously. It was necessary to study these grid effects related to TWINGB for adapting the TWINGB code to a block-structured grid.

### A. Density Calculation in TWINGB

In TWINGB, density  $\rho$  is calculated using equation (7d). It follows that:

$$\begin{aligned}
 q^2 &= \begin{Bmatrix} U \\ V \\ W \end{Bmatrix}^T \begin{Bmatrix} \phi_\xi \\ \phi_\eta \\ \phi_\zeta \end{Bmatrix} \\
 &= (A_1 \phi_\xi + A_4 \phi_\eta + A_5 \phi_\zeta) \phi_\xi \\
 &\quad + (A_4 \phi_\xi + A_2 \phi_\eta + A_6 \phi_\zeta) \phi_\eta \\
 &\quad + (A_5 \phi_\xi + A_6 \phi_\eta + A_3 \phi_\zeta) \phi_\zeta \\
 &= A_1 \phi_\xi^2 + A_2 \phi_\eta^2 + A_3 \phi_\zeta^2 \\
 &\quad + 2(A_4 \phi_\xi \phi_\eta + A_5 \phi_\xi \phi_\zeta + A_6 \phi_\eta \phi_\zeta)
 \end{aligned} \tag{39}$$

and  $A_1$  and  $A_3$  to  $A_6$  are scaled by the Jacobian determinant  $D$  in the following manner:

$$\begin{aligned}
 A_1 &= (\xi_x^2 + \xi_y^2 + \xi_z^2)/D \\
 A_2 &= \eta_y^2 \\
 A_3 &= (\zeta_x^2 + \zeta_y^2 + \zeta_z^2)/D \\
 A_4 &= (\xi_y \eta_y)/D \\
 A_5 &= (\xi_x \zeta_x + \xi_y \zeta_y + \xi_z \zeta_z)/D \\
 A_6 &= (\eta_y \zeta_y)/D
 \end{aligned} \tag{40}$$

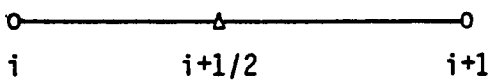
Consistent with the notation of equation (40), the formulas for computing the contravariant velocity components  $U$ ,  $V$ ,  $W$  and  $q^2$  should be written as:

$$\begin{Bmatrix} U \\ V \\ W \end{Bmatrix} = \begin{Bmatrix} DA_1 & DA_4 & DA_5 \\ DA_4 & A_2 & DA_6 \\ DA_5 & DA_6 & DA_3 \end{Bmatrix} \begin{Bmatrix} \phi_\xi \\ \phi_\eta \\ \phi_\zeta \end{Bmatrix} \tag{41}$$

$$q^2 = A_2 \phi_\eta^2 + D[A_1 \phi_\xi^2 + A_3 \phi_\zeta^2 + 2(A_4 \phi_\xi \phi_\eta + A_5 \phi_\zeta \phi_\xi + A_6 \phi_\eta \phi_\zeta)] \tag{42}$$

### 1. Approximation Using Averaging Procedures

In TWINGB, the detailed  $\rho$  computation procedure is the following:



$\phi_\xi$ ,  $\phi_\eta$ , and  $\phi_\zeta$  are computed at half  $i$  point, i.e., the  $i+1/2$  point shown in the adjacent sketch.  $A_1$  to  $A_6$  and  $D$  are computed at point  $i$  and  $i+1$ , and then averaged to obtain the corresponding values at  $i+1/2$ . Equation (42) is then used to compute  $q^2$  at  $i+1/2$ .

## 2. Approximation Using Binomial Expansions

Once  $q^2$  is computed, equation (5d) can be used to compute  $\rho$ . However, in TWINGB equation (7d) is approximated by a five-term binomial expansion. For  $\gamma = 1.4$ , equation (7d) may be written as

$$\rho = (1 - \frac{q^2}{6})^{2.5} \quad (43)$$

Define

$$Q = q^2/6 \quad (44)$$

$$\rho = (1 - Q)^{2.5} \quad (45)$$

Binomial expansion of  $\rho$  in terms of  $Q$  then leads to

$$\rho = 1 - 2.5Q + 1.875Q^2 - 0.3125Q^3 + 0.0390625Q^4 \quad (46)$$

Equation 46 offers a much faster  $\rho$  computation than equation (7d). Equation (46) is reasonably accurate for engineering application in the transonic region. Let  $\rho_{ex}$  be the exact  $\rho$  computed from equation (43) and  $\rho_a$  be the approximation solution of  $\rho$  computed by equation (46), then the relative error

$$Err = (\rho_a - \rho_{ex})/\rho_{ex} \quad (47)$$

is bounded by

$$Err < 10^{-4} \quad \text{for } M < 1.5 \quad (48)$$

where  $M$  is the local Mach number.

In order to compute  $\rho$  accurately, we must compute  $q^2$  accurately first. It is, therefore, necessary to review the accuracy of the  $q^2$  computational procedure in TWINGB, focusing on the error introduced by an arbitrary grid.

Take a uniform flow aligned with the  $x$  axis over an orthogonal grid that is aligned with the  $x$  and  $y$  axes. The grid effect can be studied by employing a 1D model in the  $x$ -direction.

## B. Effect Due to Uneven Grid Spacing

In TWINGB, for the 1D case

$$D = 1/x$$

$$U = D A_1 \phi_\xi$$

$$A_1 = 1/x$$

$$\phi_x^2 = q^2 = D A_1 \phi_\xi^2 \quad (49)$$

$$\text{Let } \phi = x \quad (50)$$

$$\text{then } (\phi_\xi)_{i+1/2} = f$$

$$(X_\xi)_i = (1 + f)/2$$

$$(X_\xi)_{i+1} = f$$

$$(A_1)_i = 2/(1+f)$$

$$(A_1)_{i+1} = 1/f$$

$$(\tilde{A}_1)_{i+1/2} = 0.5((A_1)_i + (A_1)_{i+1})$$

$$= \frac{1 + 3f}{2f(1+f)} = (\tilde{D})_{i+1/2}$$

$$(\tilde{\phi}_x^2)_{i+1/2} = [0.5(1+3f)/(1+f)]^2 \quad (51)$$

From equation (50)

$$\phi = x, \quad \phi_x = 1, \quad \phi_x^2 = 1 \text{ is the exact solution.}$$

From equation (51)  $f = 1$  implies

$$(\tilde{\phi}_x^2)_{i+1/2} = 1$$

Approximate formula recovers the freestream solution  $\phi_x^2 = 1$  when the mesh spacing is uniform. For  $f \neq 1$ , then in general

$$(\tilde{\phi}_x^2)_{i+1/2} \neq 1$$

We plot  $(\tilde{\phi}_x^2)_{i+1/2}$  against  $f$  in Figure 10 to show the effect due to the uneven grid on the result.

We also used the TWINGB code to compute the  $\rho$  corresponding to a uniform flow,  $\phi = q_\infty x$ ,  $M_\infty = .86$ , over a flat plate at zero incident angle. Here we know that the exact solutions are

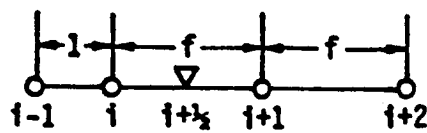
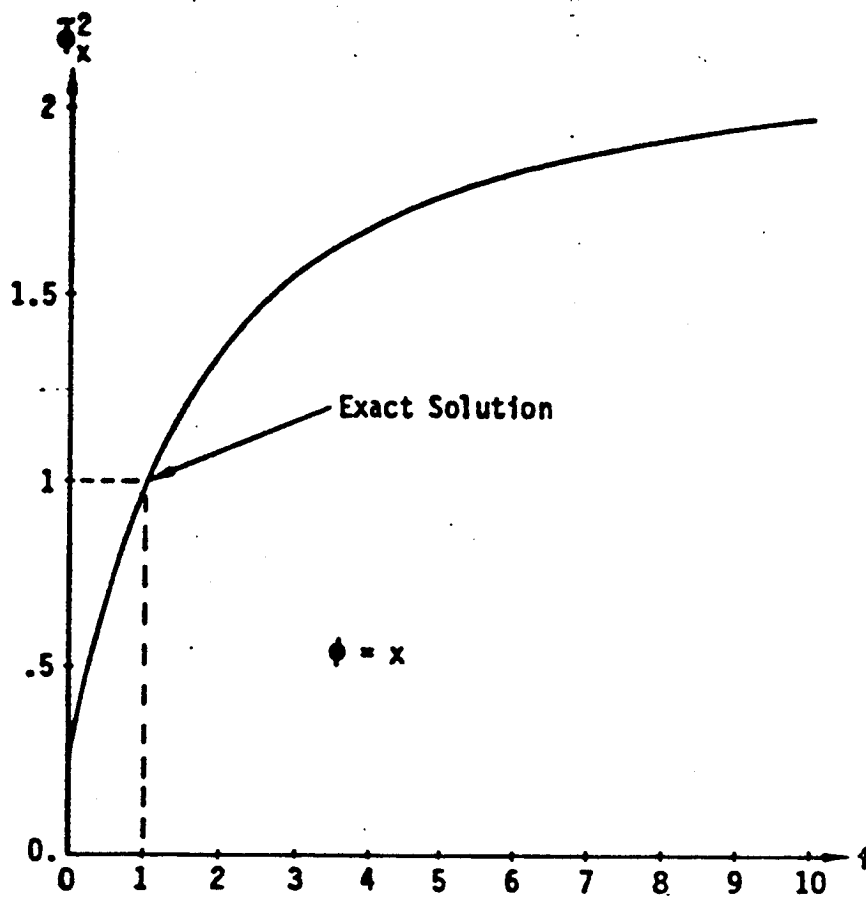
$$\phi_x = q = q_\infty, \quad \rho = \rho_\infty \quad \text{everywhere}$$

The grid layout for this computation is given in Figure 11. It should be noted that along the horizontal lines  $z = .4, .6$  and  $.8$ , the grid spacing in  $x$  jumps across the block boundary. The density computed along these three lines is depicted in Figure 12.

Along these three lines, the  $\rho$  values deviate from the freestream value  $\rho_\infty = .7083$  significantly near the block boundary. The error in  $\rho$  is resulting from the error in the  $\phi_x^2$  computation on an uneven grid (Figure 12). At all other points, the  $\rho$  computed is close to the freestream value. The error in  $\rho$  near the block boundary is so severe that some points  $\rho$  is smaller than its critical value  $\rho^* = .634$ .

### C. Effect Due to Grid Distortion

Here we want to examine the effect due to grid distortion. The test problem we chose here is that of a uniform flow around a circular cylinder. We use a smooth grid depicted in Figure 13 to produce the baseline solutions. This grid has 60 cells in the circumferential direction with uniform spacing and 10 cells in the radial direction. In the radial direction, the grid spacing is stretched from the cylinder surface to the farfield with a stretching ratio  $FR = 1.2$ . The farfield boundary is located at 5 radii from the cylinder center. The grid in Figure 13 is then distorted by redistributing the grid points along a vertical grid line  $FR = 1.3$  for this one line (Figure 14).



$\tau_x^2$  Computed at  $i+\frac{1}{2}$

Figure 10. Grid Effect on  $\tau_x^2$  Value Computed in TWINGB.



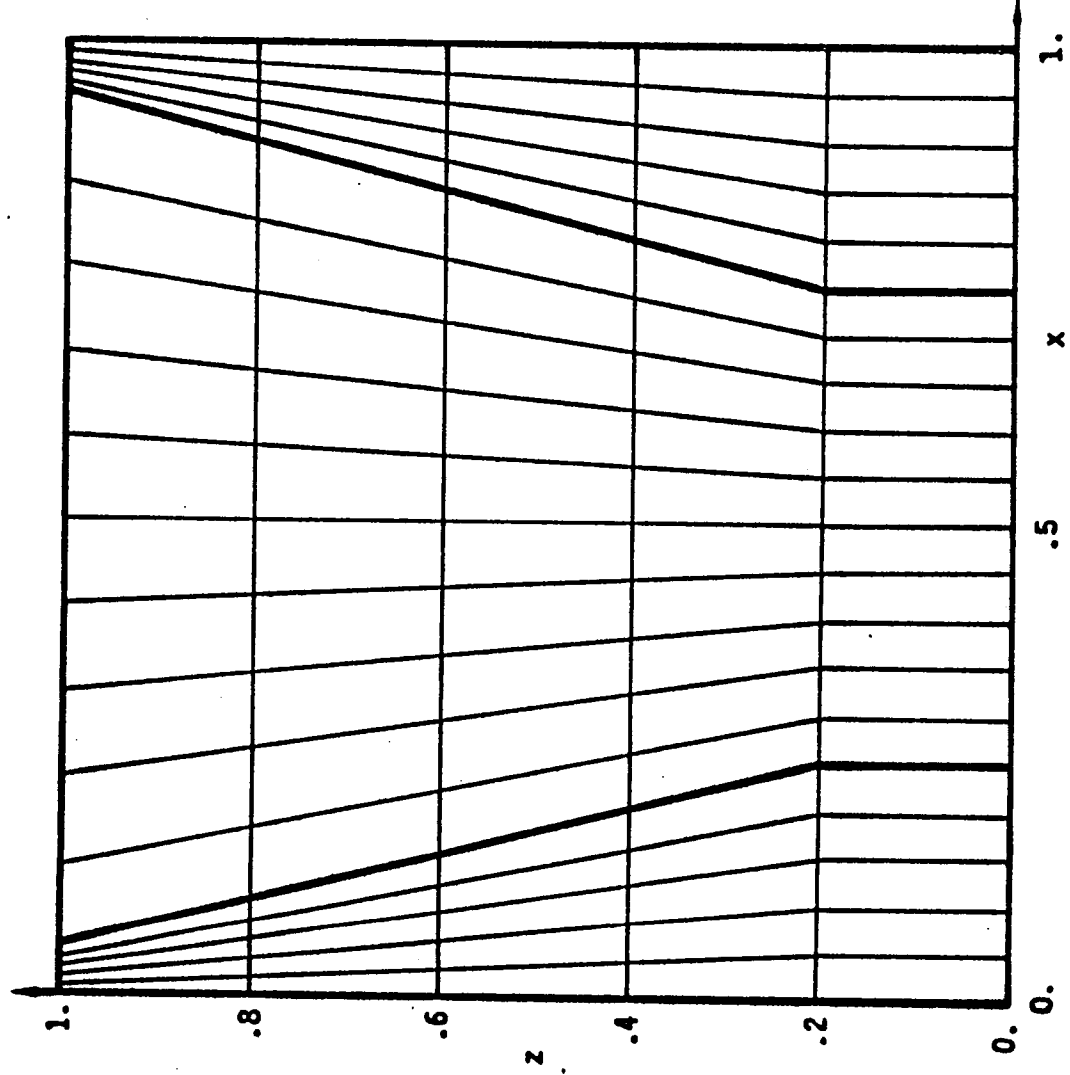


Figure 11. A Simple Grid to Test The Grid Sensitivity of TWINGB.

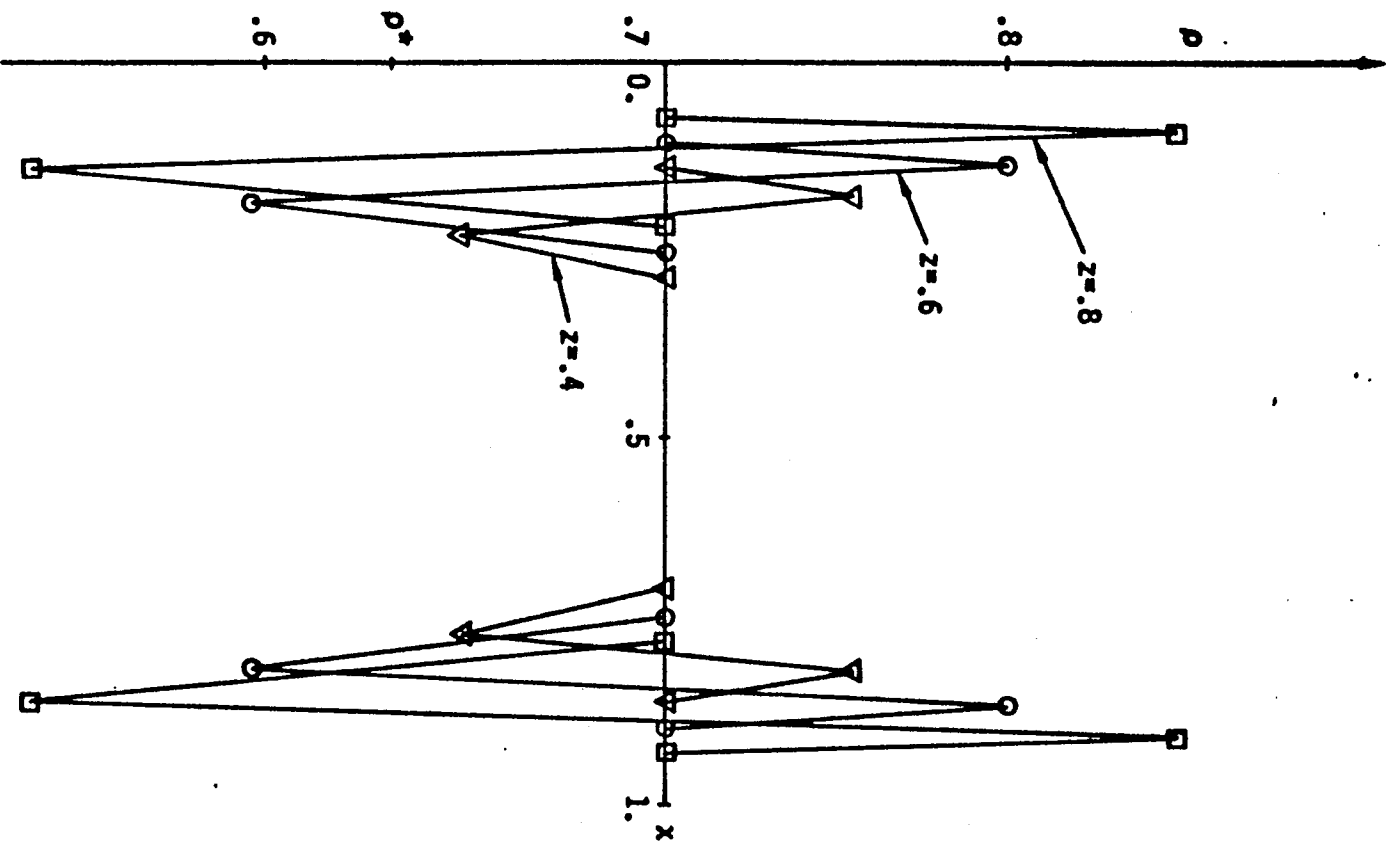


Figure 12. Error in Density Due to Grid Effect.

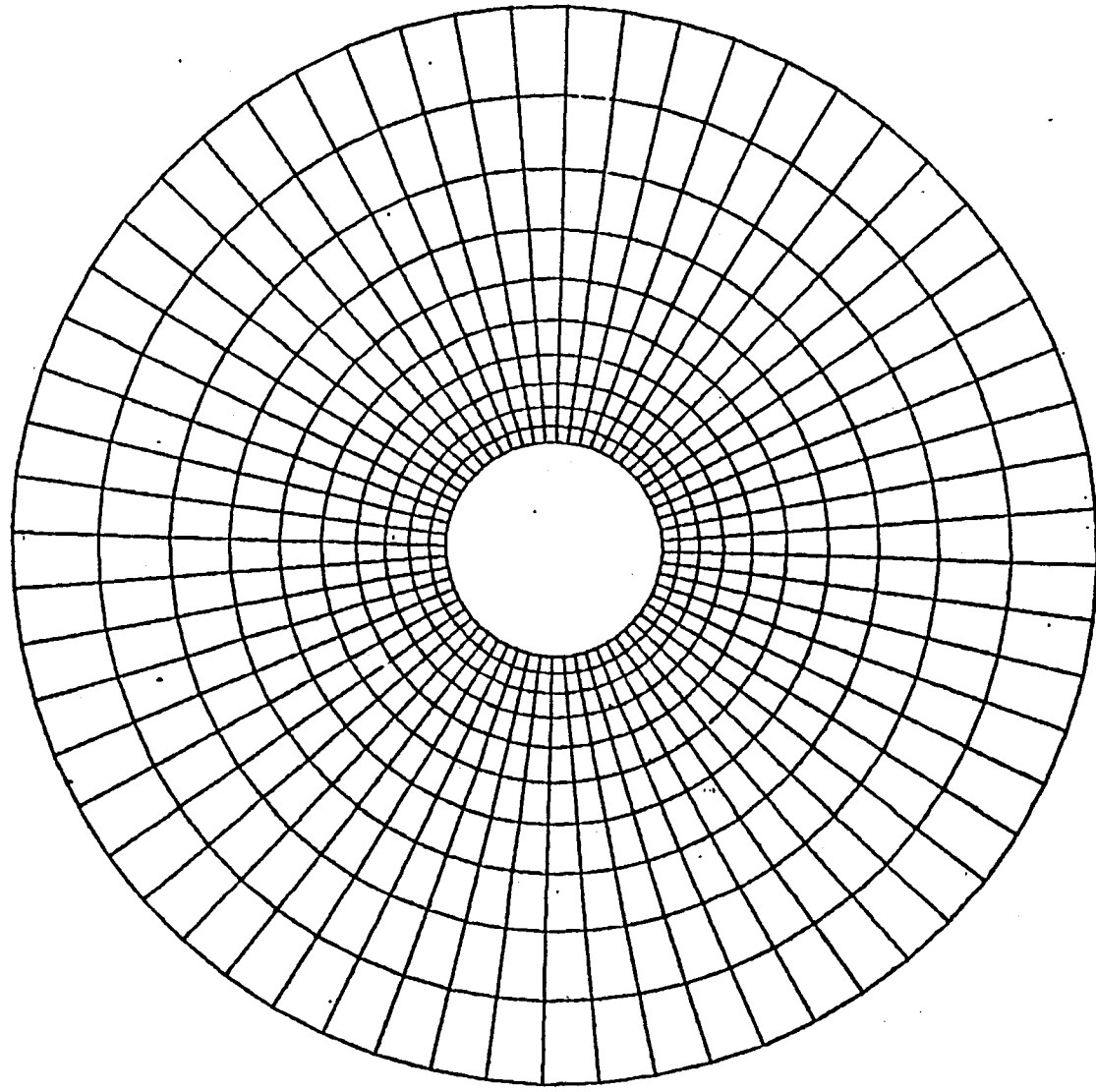


Figure 13. A Smooth Grid for Circular Cylinder Case,  $FR=1.2$ .

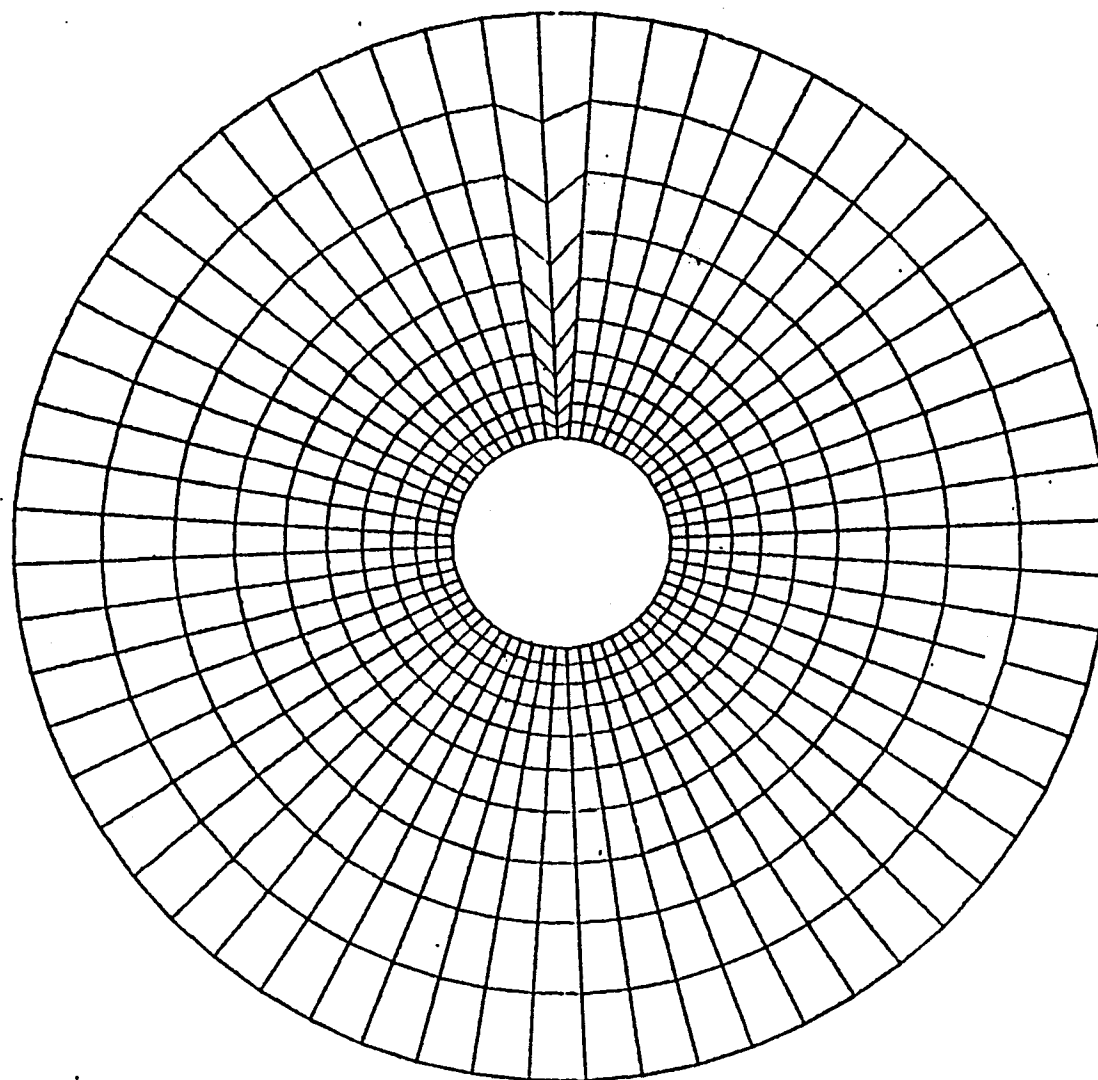


Figure 14. A Locally Distorted Grid for Circular Cylinder Case,  $FR=1.2$ .

Comparison of local Mach numbers on the cylinder surface for the baseline grid and the distorted grid are given in Figure 15 for freestream Mach number:  $M_\infty = .45$  and in Figure 16 for  $M_\infty = .01$ . The solutions are severely affected near the location where the grid is distorted in Figure 15. This effect exists even for the case of nearly incompressible flow with  $M_\infty = .01$  (Figure 16).

#### D. Effect Due to Grid Aspect Ratio

Block-structured grid sometimes results in some grid cells with high aspect ratio. It was therefore necessary to study the sensitivity to grid aspect ratio for the basic TWINGB code. For the cylinder flow, we altered the baseline grid of Figure 13 by increasing the grid spacing stretching ratio FR to 1.5.

The resulting grid is given in Figure 17 and numerical experiments produced divergent results from TWINGB for both  $M_\infty = .45$  and  $.01$ . This suggests that the stability of the flow solution algorithm in TWINGB can be severely affected by the presence of high aspect ratio grid cells.

#### E. Conclusions on Grid Effect Study

Numerical experiments showed that the inconsistent mapping procedures in TWINGB lead to significant error that is proportional to the nonuniformity of the grid. The error induced can be severe in the presence of local grid distortion. This error can cause instability in the iterative process when the grid aspect ratio is high enough. A summary of the TWINGB grid effect study can be seen in Table 2.

This error can be eliminated by using a consistent mapping procedure. The difference operators for differencing  $\phi$  and for generating the Jacobian matrix should be the same to ensure accuracy in the arbitrary grid. E.g., in the 1D case, all difference formulas should be centered at the Half-I point.

The extension of this concept to 3D is straightforward if we follow the concept of isoparametric mapping taking a suitable subdomain definition (see Appendix A for more detail). Refer to Figure 18, inside the subdomain,  $\phi$  and  $x, y, z$  can be locally approximated by a consistent polynomial approximation.

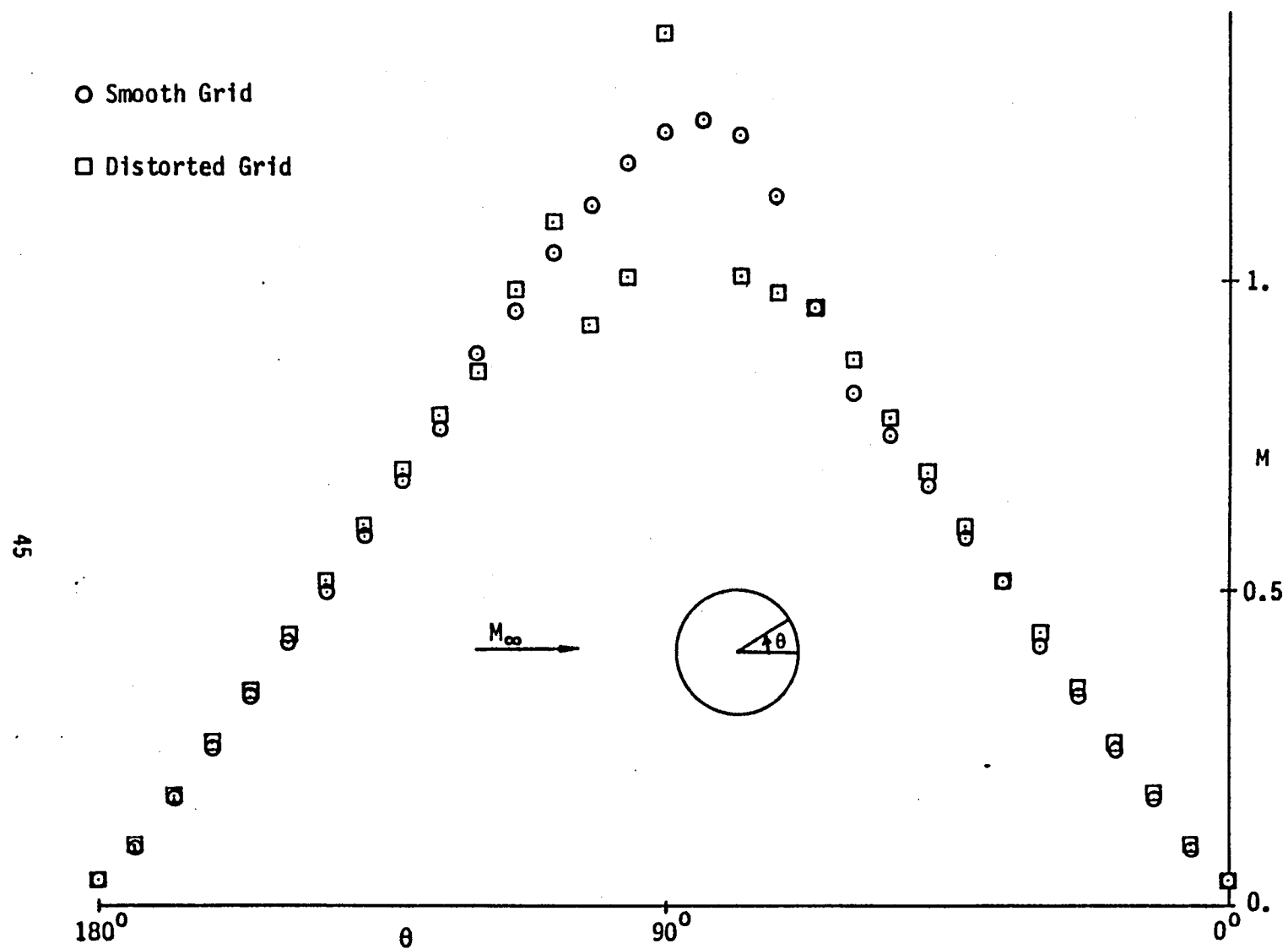


Figure 15. Grid Effect Study. Circular Cylinder.  $M_\infty=0.45$ , Upper Surface  $M$ .

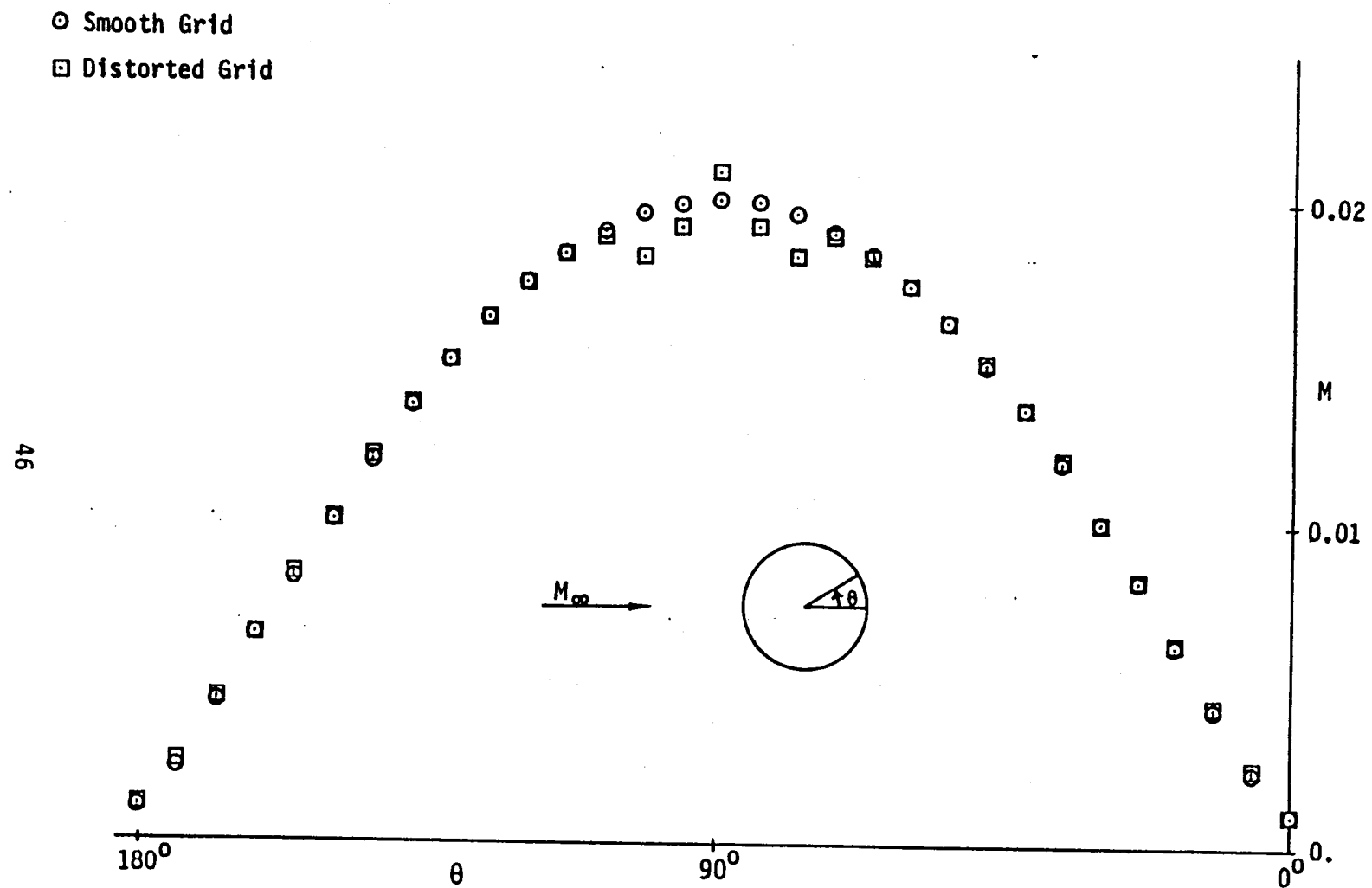


Figure 16. Grid Effect Study, Circular Cylinder,  $M_\infty = 0.01$ , Upper Surface  $M$ .

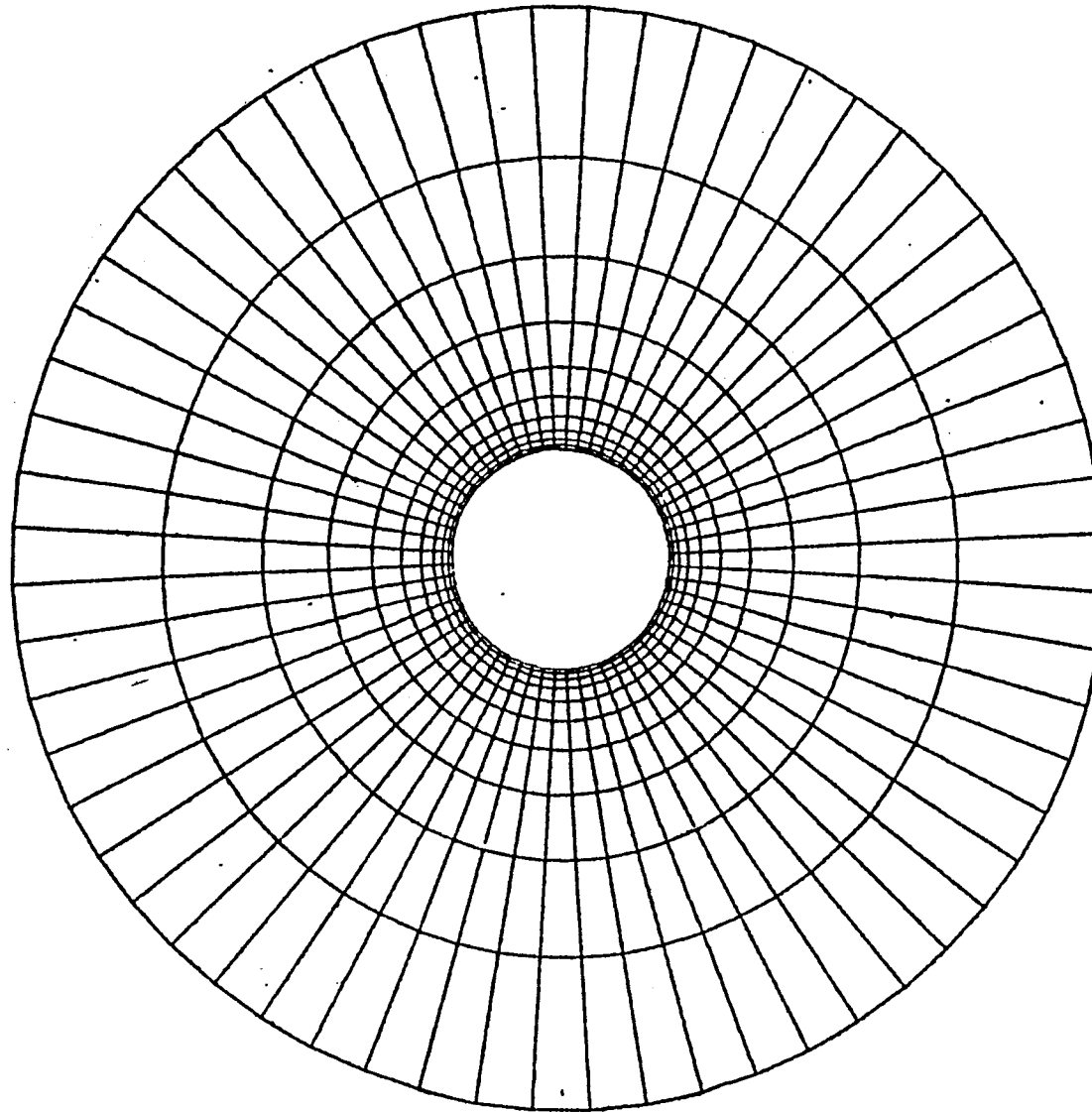


Figure 17. A Smooth Grid for Circular Cylinder Case,  $FR=1.5$ .



Table 2 TWINGB Grid Effect Study Summary

Test	Flow Type	Discussions
Grid spacing discontinuity	High subsonic	Discontinuity in the results
Local Grid distortion	Transonic	Very poor results locally
Local Grid distortion	Low subsonic	Poor results locally
High Grid Aspect Ratio	Transonic	Diverged
High Grid Aspect Ratio	Low subsonic	Diverged

Let  $f$  be representing either  $\phi$  or  $x, y, z$ , then a suitable expansion formula for the computational element in Figure 18 may be written as

$$f = \sum_i \sum_j \sum_k N_i M_j L_k f_{i,j,k} \quad (52a)$$

$$N_1 = \xi - 0.5 \quad N_2 = \xi + 0.5$$

$$M_1 = 0.5 \eta (\eta - 1) \quad M_2 = 1 - \eta^2 \quad M_3 = 0.5 \eta (\eta + 1)$$

$$L_1 = 0.5 \zeta (\zeta - 1) \quad L_2 = 1 - \zeta^2 \quad L_3 = 0.5 \zeta (\zeta + 1) \quad (52b)$$

and the computational element is defined by

$$-0.5 \leq \xi \leq 0.5 \quad -1.0 \leq \eta \leq 1.0 \quad -1.0 \leq \zeta \leq 1.0 \quad (52c)$$

The consistent difference formulas for  $\phi$  and  $x, y, z$  can be obtained by differentiating the polynomial approximations and taking the cell center

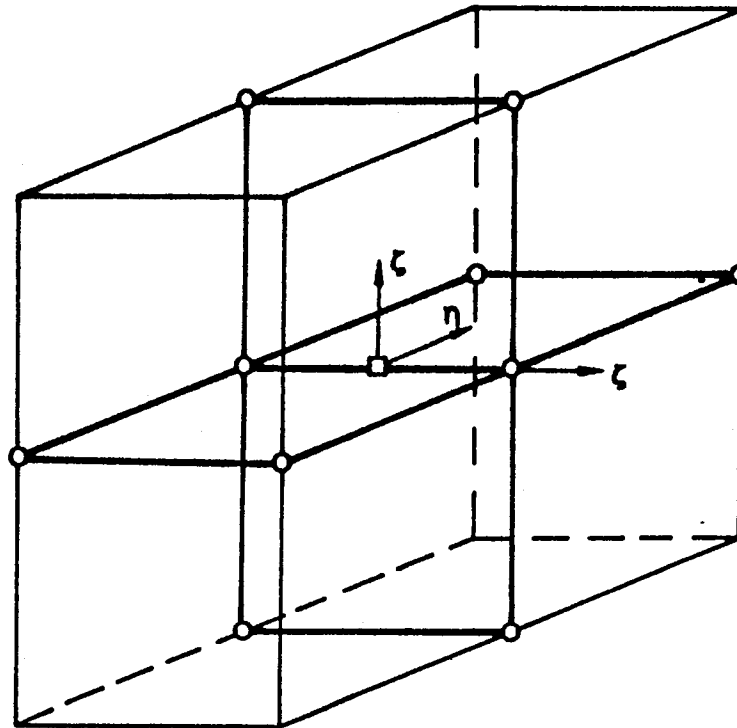


Figure 18. Isoparametric Mapping and Subdomain Definition

$\xi = \eta = \zeta = 0$  as the referencing point for evaluating the difference formulas. This improved method can be used in the flow solution algorithm to extend TWINGB more effectively to flow problems involving arbitrary grids with block structure.

## VI. RESULTS FROM ADAPTATION OF TWINGB TO BLOCK-STRUCTURED GRIDS

The study of grid effects on TWINGB indicated limitations of TWINGB for arbitrary grid. The modified TWINGB (structured for block-structured grid) was tested on cases with a typical parabolic arc airfoil and wing using three-block grids. These numerical experiments identified the type of problems that can be treated by the current flow solution algorithm in TWINGB, and suggested that the current algorithm should be improved using a consistent mapping procedure (Appendix A).

The modified TWINGB code was first tested in a 2D mode to solve the airfoil problems. The 2D flow was simulated by placing a nonlifting rectangular wing between two parallel walls extended to infinity. The wing is non-twisted, and consists of same section geometries for all spanwise stations. There are five computational stations in the spanwise direction and these stations are covered by the same 2D grid. Connecting the corresponding grid points between all neighboring stations then produces a 3D grid. The results computed between stations were also the same. Only one station is needed to show the wing section geometries, grid layout, and surface pressure results for a particular test case.

We then test the modified TWINGB code in a 3D mode on wing problems. For this study, the rectangular wing planform in the 2D case was changed by adding sweep to the planform. This generates a swept non-tapered wing between two parallel walls extended to infinity. To study the 3D effects, including the variations of the solution in the spanwise direction, 20 computational stations were used in this direction. For simplicity, each station is still covered by the same 2D grid.

Unless otherwise stated, the parameters for AF2 iterations are set at default values. In the default mode, the time-like dissipation coefficient  $\beta_t = .1$ ,  $\beta_\eta = 0$ , the artificial viscosity coefficient  $C = 1.2$ , the relaxation factor  $\omega = 1.8$ , the  $\alpha$  sequence end points are ( $\alpha_L = 0.4$ ,  $\alpha_H = 4$ ), and there are eight elements in the  $\alpha$  sequence.

All TWINGB calculations for the 2D cases use a mesh with 6,000 cells ( $80 \times 5 \times 15$ ). This places 40 cells from leading to trailing edge. The TWINGB calculations for the 3D cases use a mesh with 22,800 cells ( $80 \times 19 \times 15$ ). The grid spacing in the spanwise direction (y direction) is uniform for both 2D and 3D cases.

#### A. Test Cases for Airfoil Problems

##### 1. Uniform Flow Over a Flat Plate

For this test case, the grid in each spanwise station has 80 cells in the x direction (streamwise direction) and 15 cells in the z direction (normal to the wing direction). For a grid with equal spacing in x and equal spacing in z (Figure 19), numerical experiments at  $M_\infty = .9$  showed that the AF2 iteration is stable and that it completely recovers the freestream flow. Only the lower half space is needed since the flow is symmetric with respect to  $z = 0$ . The lower half space (instead of the upper half space) should be used to make the  $\xi$ ,  $\eta$ , and  $\zeta$  coordinate system a right hand system - the same system as the x, y, and z coordinate system. The use of inconsistent coordinate systems would lead to numerical instability in the AF2 iteration beyond the simple case of two dimensional totally subsonic flow.

We next stretched the grid spacing in the z direction with a stretching ratio  $FZ = 1.2$ . The resulting grid is depicted in Figure 20. This grid leads to numerical instability with TWINGB for both  $M_\infty = .9$  and .0001. The maximum residual  $R_{\max}$  and maximum correction  $C_{\max}$  versus iteration is plotted in Figure 21 for the case of  $M_\infty = .0001$ . Clearly, the iteration is diverging. The peaks in the AF2 convergence (or divergence) history correspond to the small value of  $\alpha$  (i.e., larger pseudo-time step) in the eight element sequence.

We then reduced FZ to 1.1 (Figure 22), and the resulting AF2 iteration for the flow solution becomes stable (for the case of  $M_\infty = .9$ ) and the solution recovers the freestream flow.

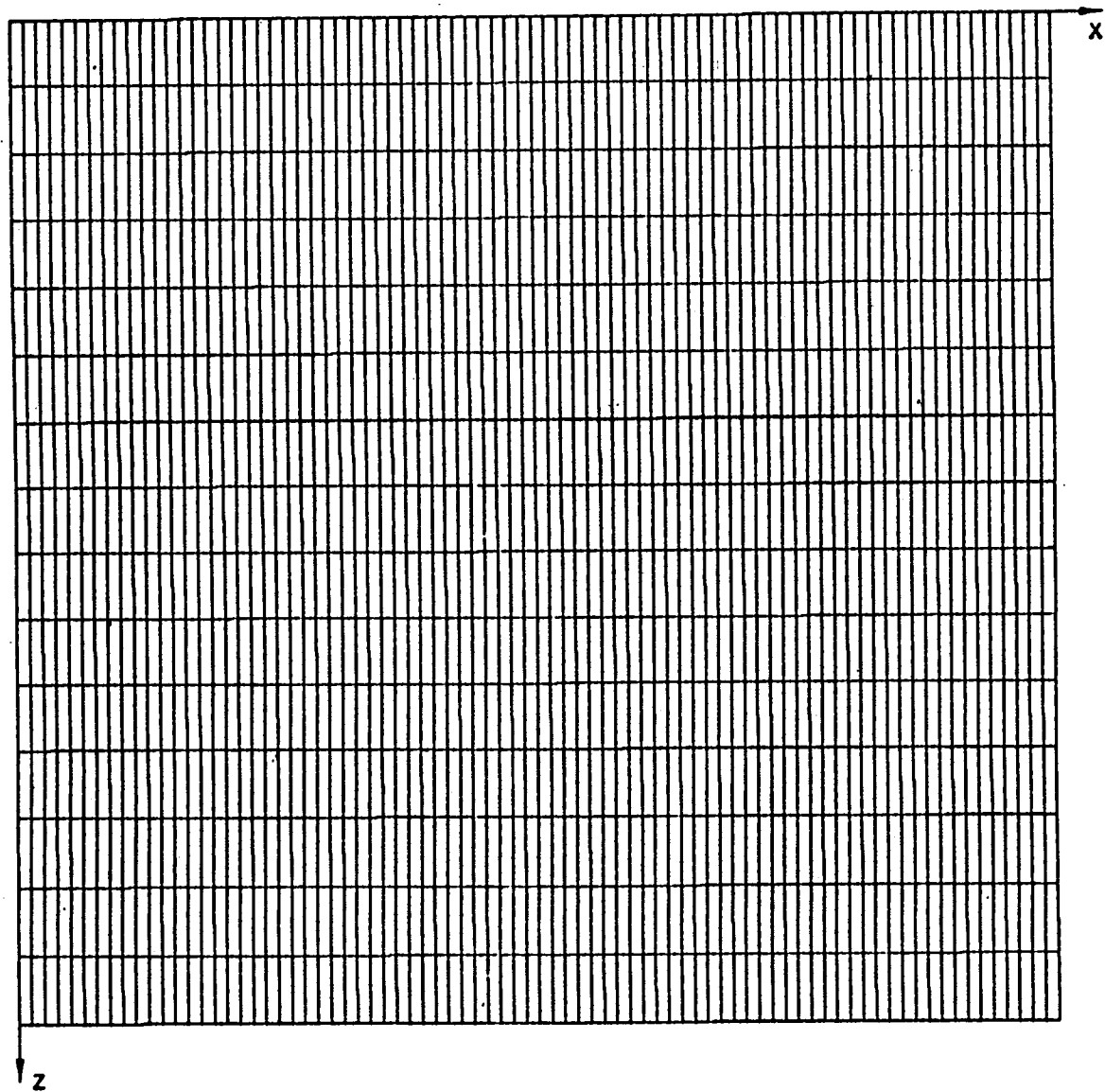


Figure 19. Grid-A, Flat Plate, Uniform Grid.

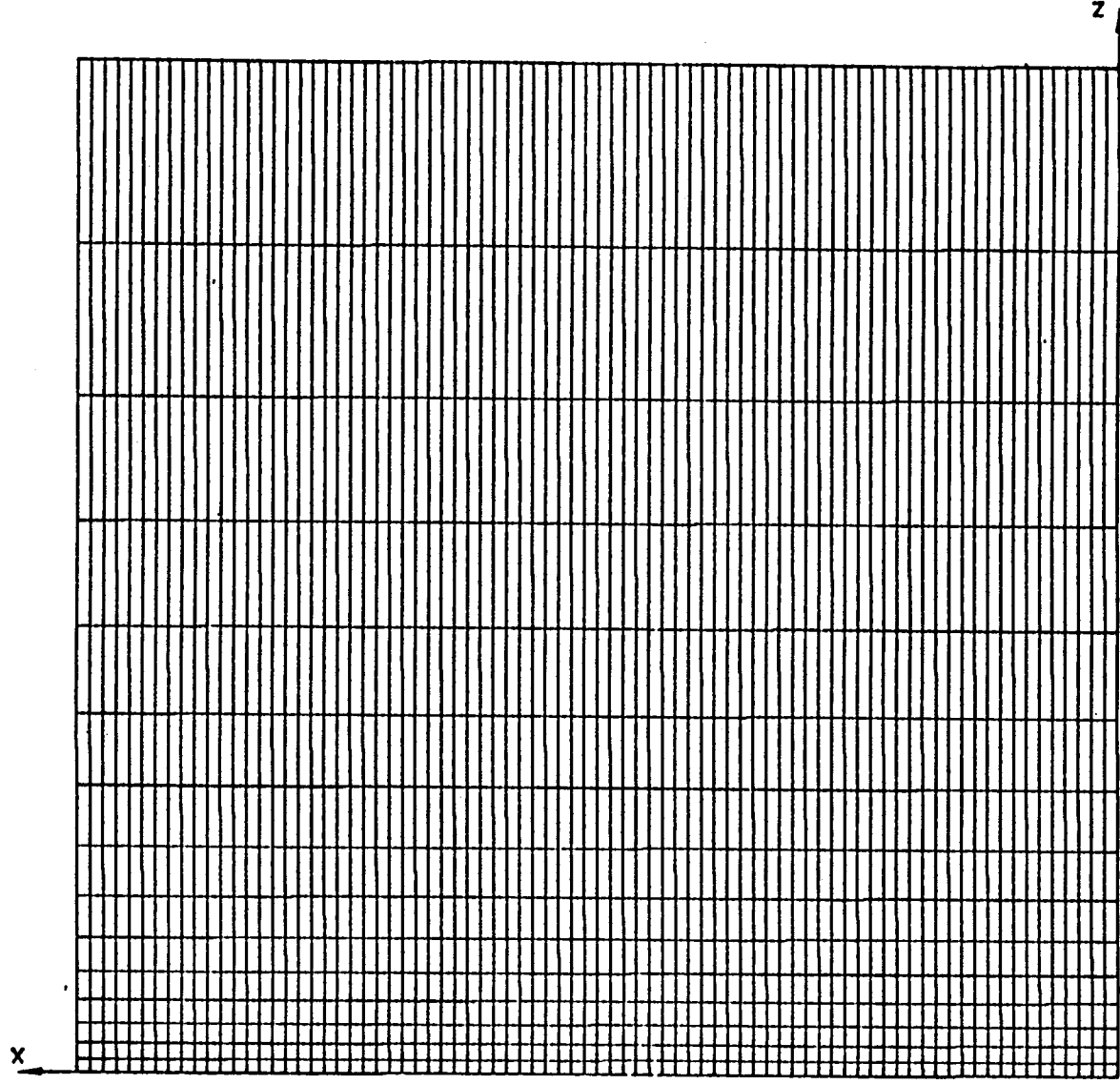


Figure 20. Grid-B, Flat Plate,  $FZ=1.2$ .

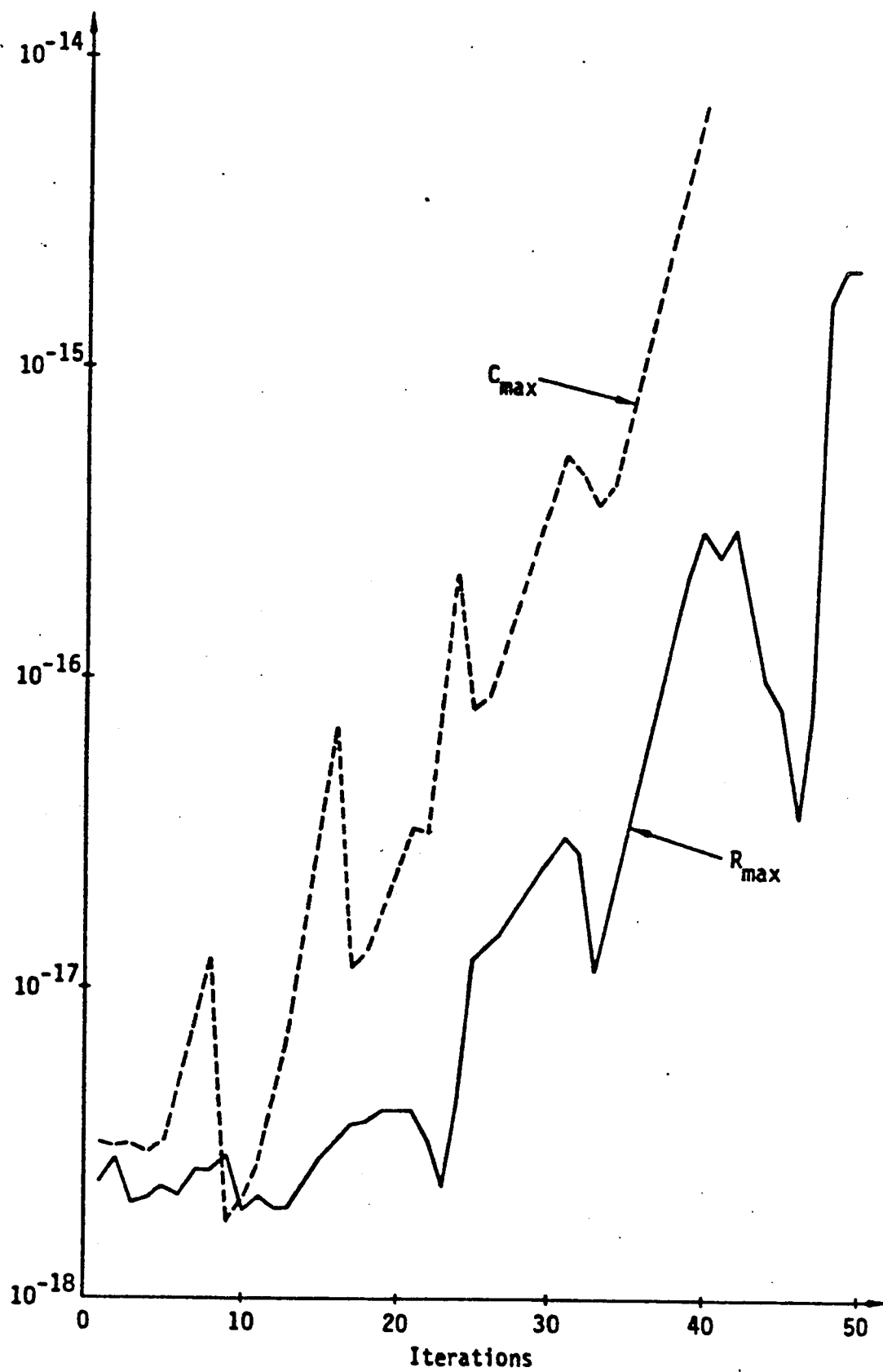


Figure 21. Growth of  $R_{max}$  and  $C_{max}$  for Flow Over a Flat Plate, Grid-B,  $M_\infty = .0001$ .



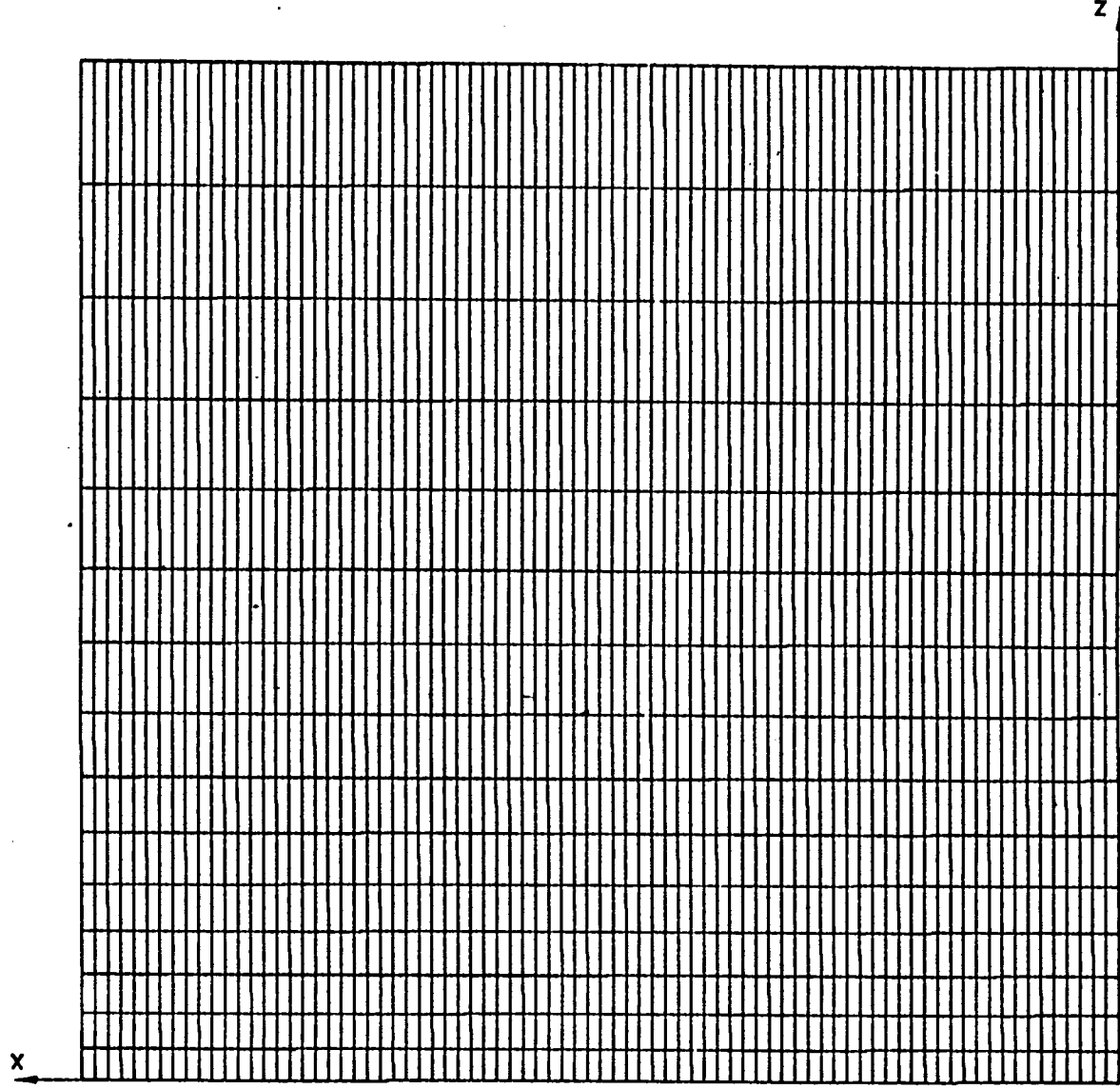


Figure 22. Grid-C, Flat Plate,  $FZ=1.1$ .

We next generated a three-block grid keeping  $FZ=1.1$  (Figure 23). The first block (left hand block) contains  $i = 1$  to 21, the second block (central block)  $i = 21$  to 61, and the third block (right hand block)  $i = 61$  to 81. For the central block, uniform grid spacing in  $x$  is used. For the left hand and right hand block, the  $x$  grid spacing was stretched with a stretching ratio  $FX = 1.125$ . The block boundaries are located at  $x = 0$  ( $i = 1$ ),  $x = .4056$  ( $i = 21$ ),  $x = .5944$  ( $i = 61$ ), and  $x = 1$  ( $i = 81$ ). The flat plate is placed at  $z = 0$  from  $x = .4056$  to  $x = .5944$  with a chord length of .1888. The far field in the  $z$  direction is at  $z = -1$ . With this grid, the AF2 iteration is stable and the convergence histories of  $R_{\max}$  and  $C_{\max}$  are given in Figure 24 for  $M_{\infty} = .9$  and in Figure 25 for  $M_{\infty} = .0001$ .

The exact solution for this case is  $C_p = 0$  everywhere. To check the solution accuracy based on the grid in Figure 23, the pressure coefficient  $C_p$  on the flat plate surface is plotted against the normalized  $x$  coordinates (Figure 26) for the case of  $M_{\infty} = .90$ . The resulting  $C_p$  is very small but not zero. This illustrated the grid effects on the AF2 flow solution algorithm. This test case is simple and the relevant results appear to be acceptable within engineering accuracy. It remains to be examined what grid effects will do for more difficult cases.

## 2. 0.03 Thick Parabolic Arc Airfoil

The three-block grid of Figure 23 can be perturbed slightly to produce a grid for the case of flow over a thin parabolic arc airfoil. A grid so produced (Grid-E) is depicted in Figure 27 for a 0.03 thick parabolic arc airfoil. The freestream Mach number  $M_{\infty} = .75$  and the convergence history for  $R_{\max}$  and  $C_{\max}$  are shown in Figure 28. Rapid reduction of  $R_{\max}$  and  $C_{\max}$  are realized showing the effectiveness of the AF2 iteration for this test case. The  $R_{\max}$ , which is a measure of how the governing equation is satisfied, drops three order of magnitudes in 37 iterations.

The surface  $C_p$  is shown in Figure 29 (square symbol). In the same figure, the  $C_p$  computed from TCAS is given by the solid line for comparison. The TCAS code was modified from TAIR code by Kwak (private communications at NASA Ames). Both codes produce symmetrical results - a correct flow behavior for

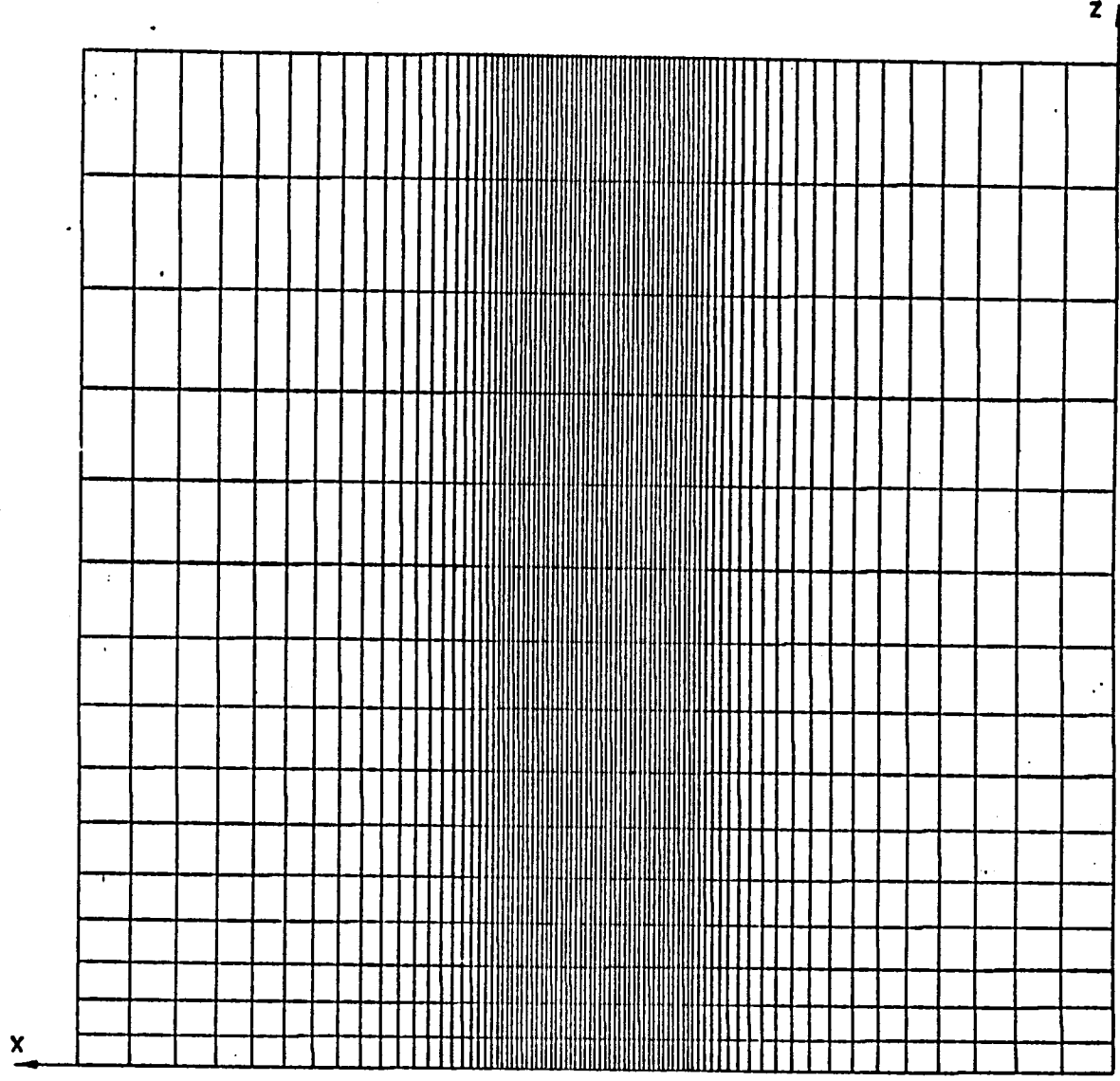


Figure 23. Grid-D, Flat Plate,  $FX=1.125$ ,  $FZ=1.1$ .

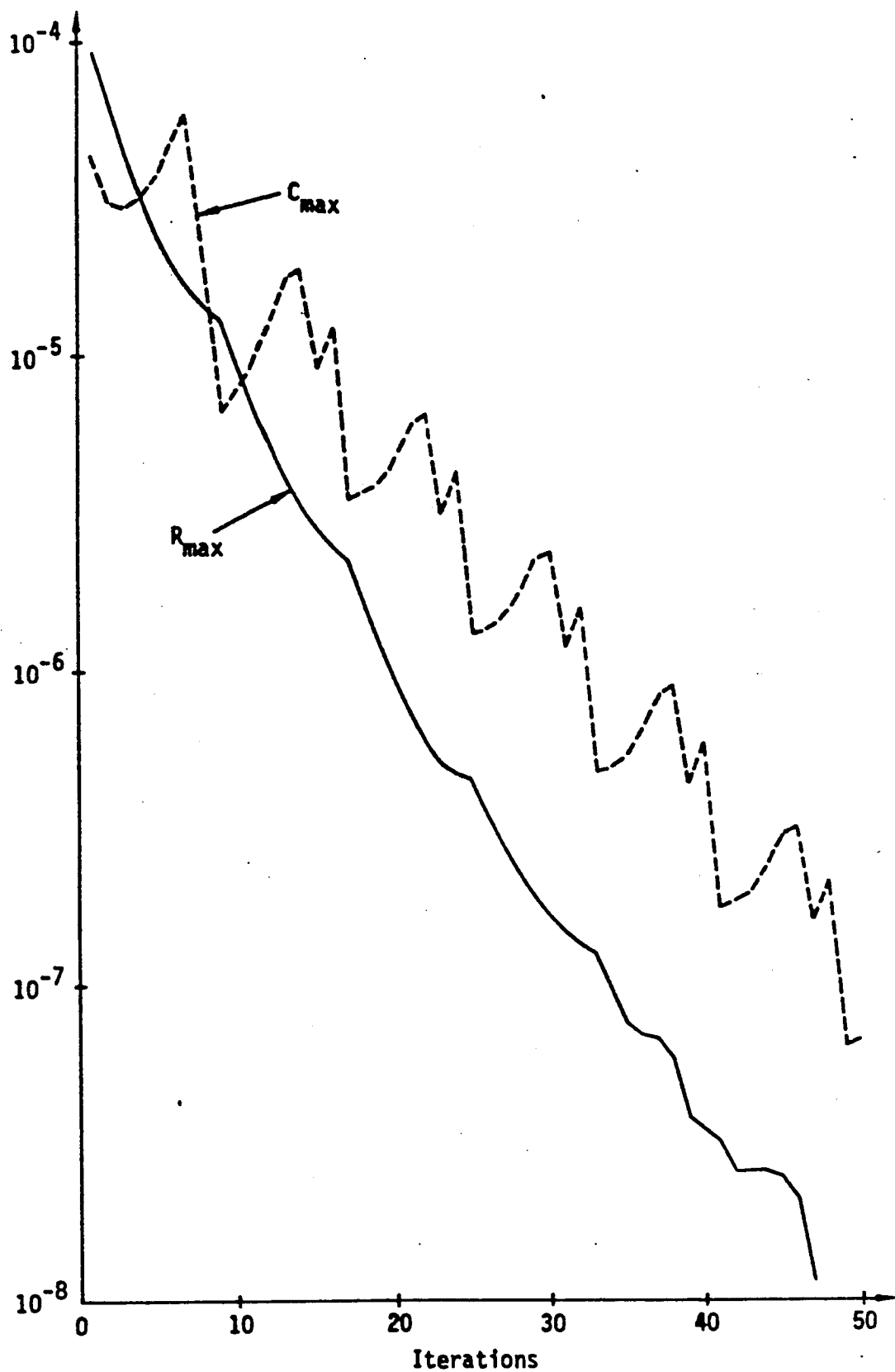


Figure 24. Convergence Histories for Flow Over a Flat Plate, Grid-D,  $M_\infty = .9$ .

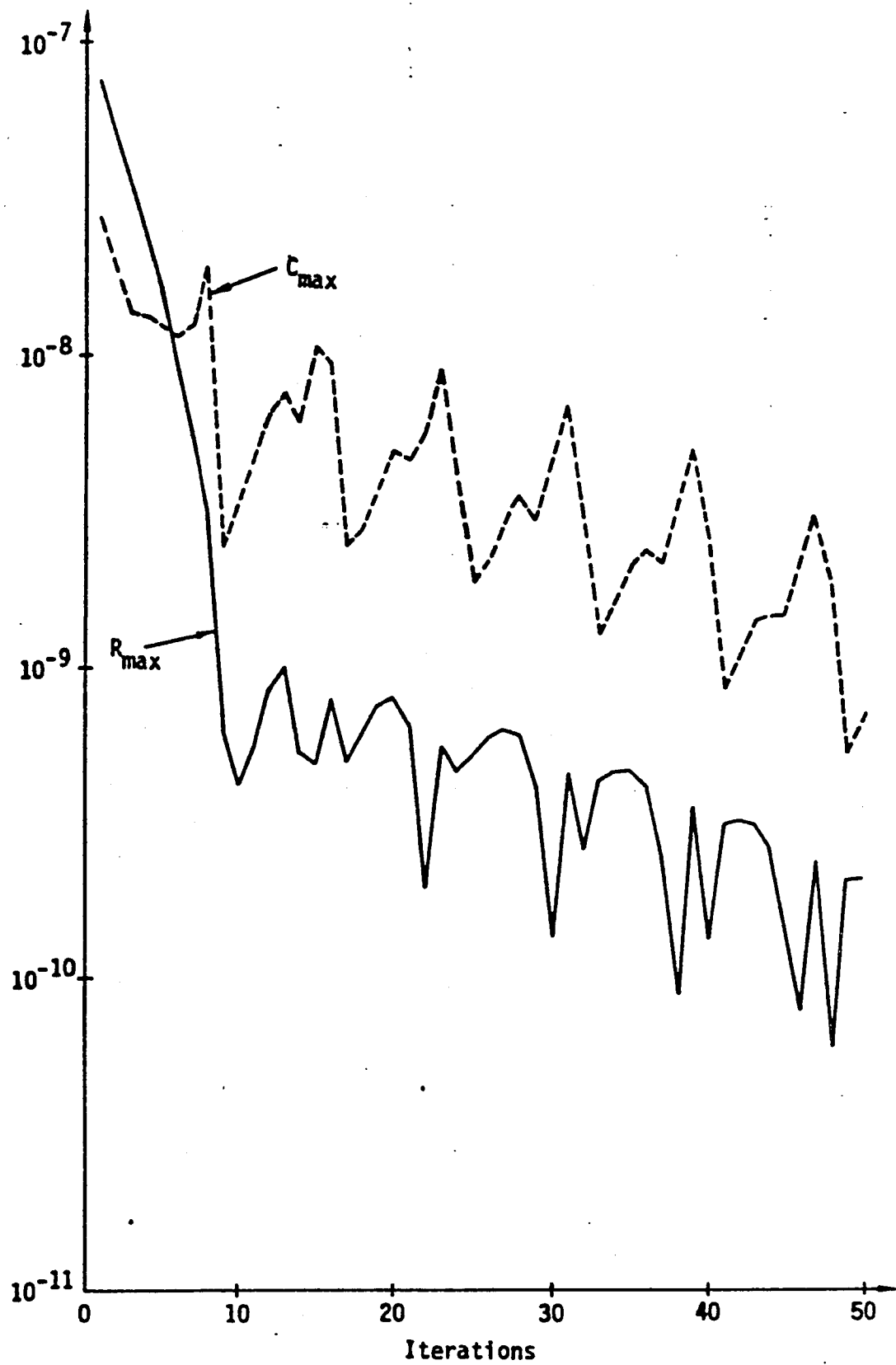


Figure 25. Convergence Histories for Flow Over a Flat Plate, Grid-D,  $M_{\infty}=.0001$ .

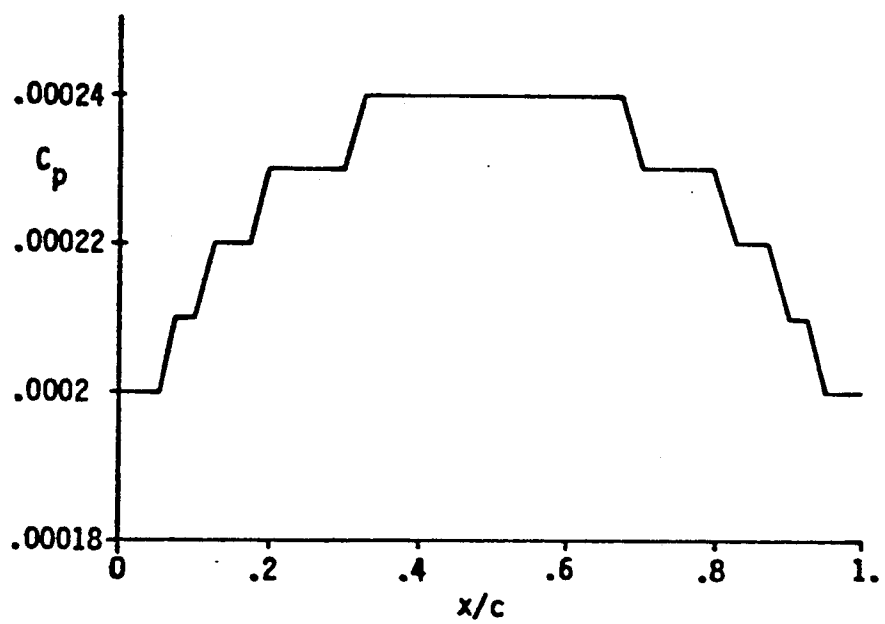


Figure 26. Nonzero  $C_p$  on the Flat Plate Surface Due to Grid Effect, Grid-D.

2D subcritical flow over a symmetrical airfoil. The general agreement between the two results is good. The differences in the two solutions are mainly caused by two factors. Firstly, different grids are used; specifically, TCAS uses many more grid cells in the  $z$  direction. Secondly, some detail in the solution procedures are different between the two codes (e.g.,  $\rho$  is computed at cell centers for TCAS but computed at half- $I$  points for TWINGB).

### 3. 0.12 Thick Parabolic Arc Airfoil

The grid of Figure 27 can be further perturbed to produce a grid for flows over a 0.12 thickness parabolic arc airfoil (Grid-F, Figure 30). Grid-F was used for calculating the transonic case of  $M_\infty = .8$ .

Increasing the airfoil thickness produces stronger perturbation to the uniform flow. The flow becomes transonic for this combination of airfoil thickness and freestream Mach number. The solution process for this case was divergent. The growth of  $R_{\max}$  and  $C_{\max}$  are given in Figure 31 for  $M_\infty = .8$ . The AF2 iteration appears to be slowly divergent in an oscillating manner. Adjustment of parameters in the AF2 iteration seems to be ineffective in getting the solution to converge.

The growth of  $R_{\max}$  and  $C_{\max}$  are given in Figures 32 to 35 to study the parameter variations. Figure 32 is the case for  $\omega = 1$ , apparently the solution is not convergent. We then increased the magnitude of the  $\alpha$  sequence end points to  $\alpha_L = 1$  and  $\alpha_H = 20$ . This has the effect of reducing the time-like step to improve the numerical stability. Even with such an  $\alpha$  sequence, the solution is still not convergent. We next increased the time-like dissipation coefficient  $\beta_\epsilon$  from .1 to 1 and add it everywhere in the flow field. The solution is still not convergent (Figure 34). The numerical instability appears to be caused by two major effects, the transonic effect and the grid effect. In order to solve this case, we need to study each effect separately.

These effects were studied individually as follows. Using the Grid-F for the case of  $M_\infty = .0001$ , the convergence histories are given in Figure 35 where

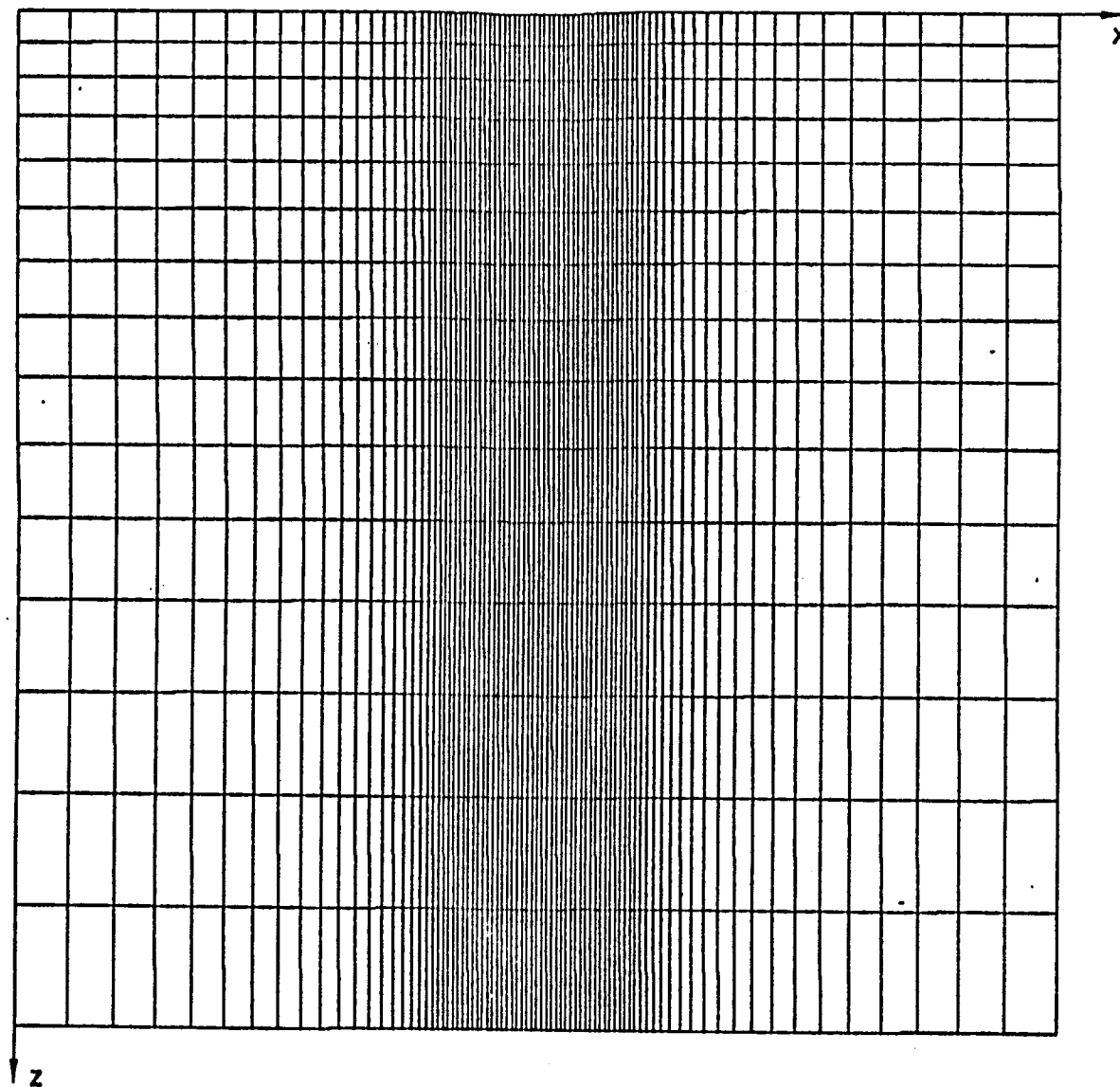


Figure 27. Grid-E, .03 Parabolic Arc Airfoil,  $FX=1.125$ ;  $FZ=1.1$ .



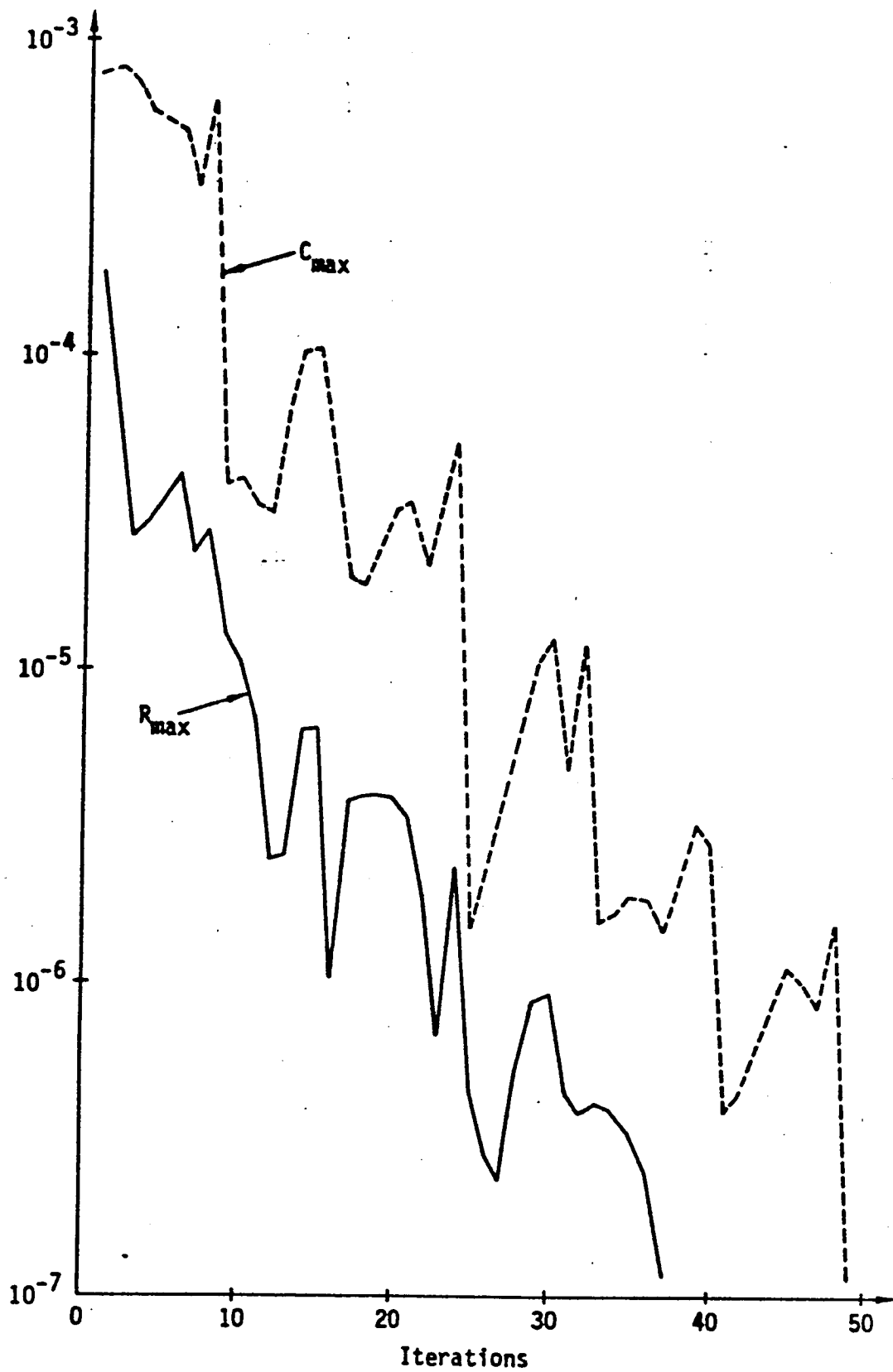


Figure 28. Convergence Histories for Flow Over a .03 Parabolic Arc Airfoil, Grid-E,  $M_\infty = .75$ .

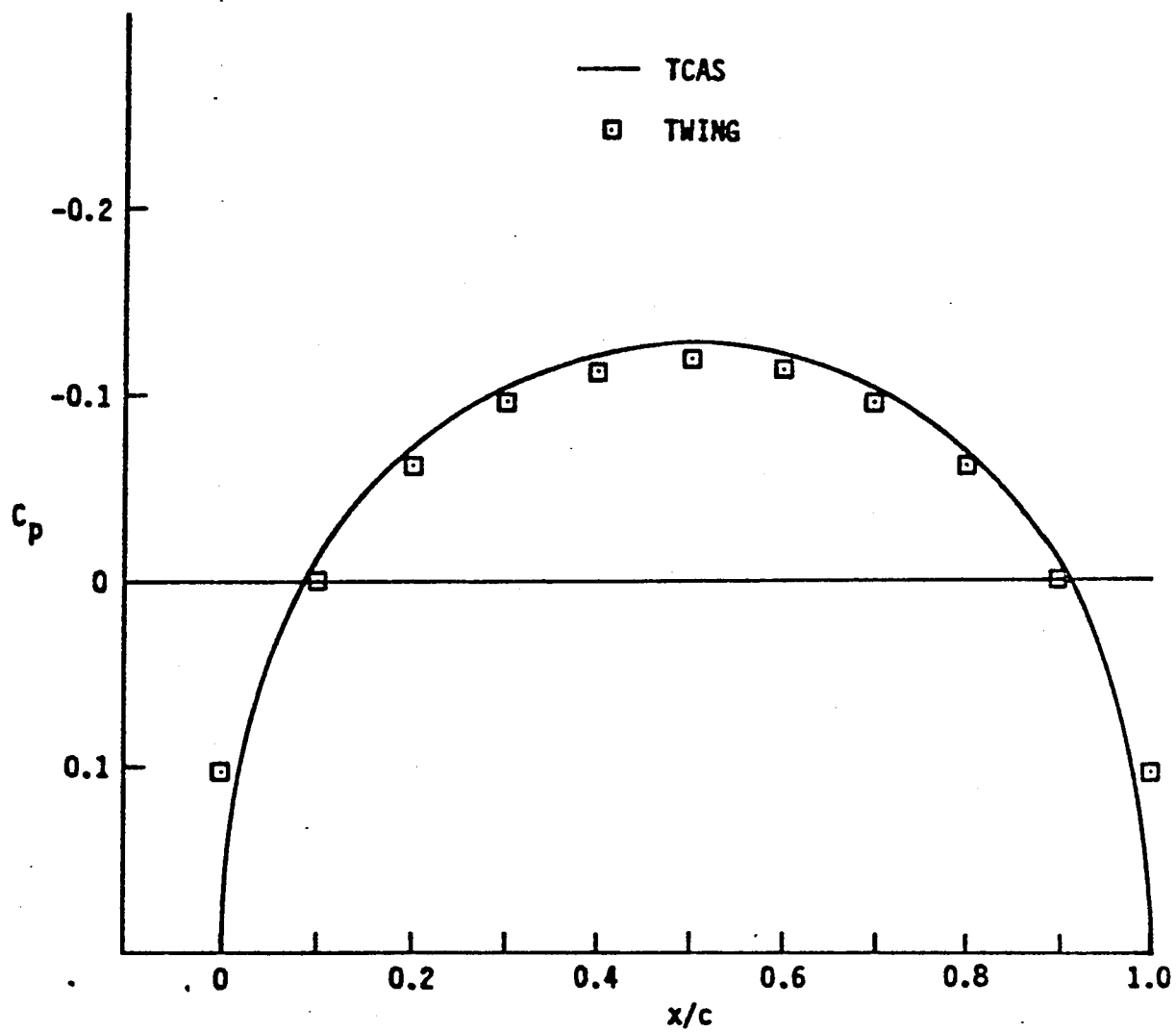


Figure 29. Surface  $C_p$  for Flow Over a .03 Parabolic Arc Airfoil, Grid-E,  $M = .75$ .

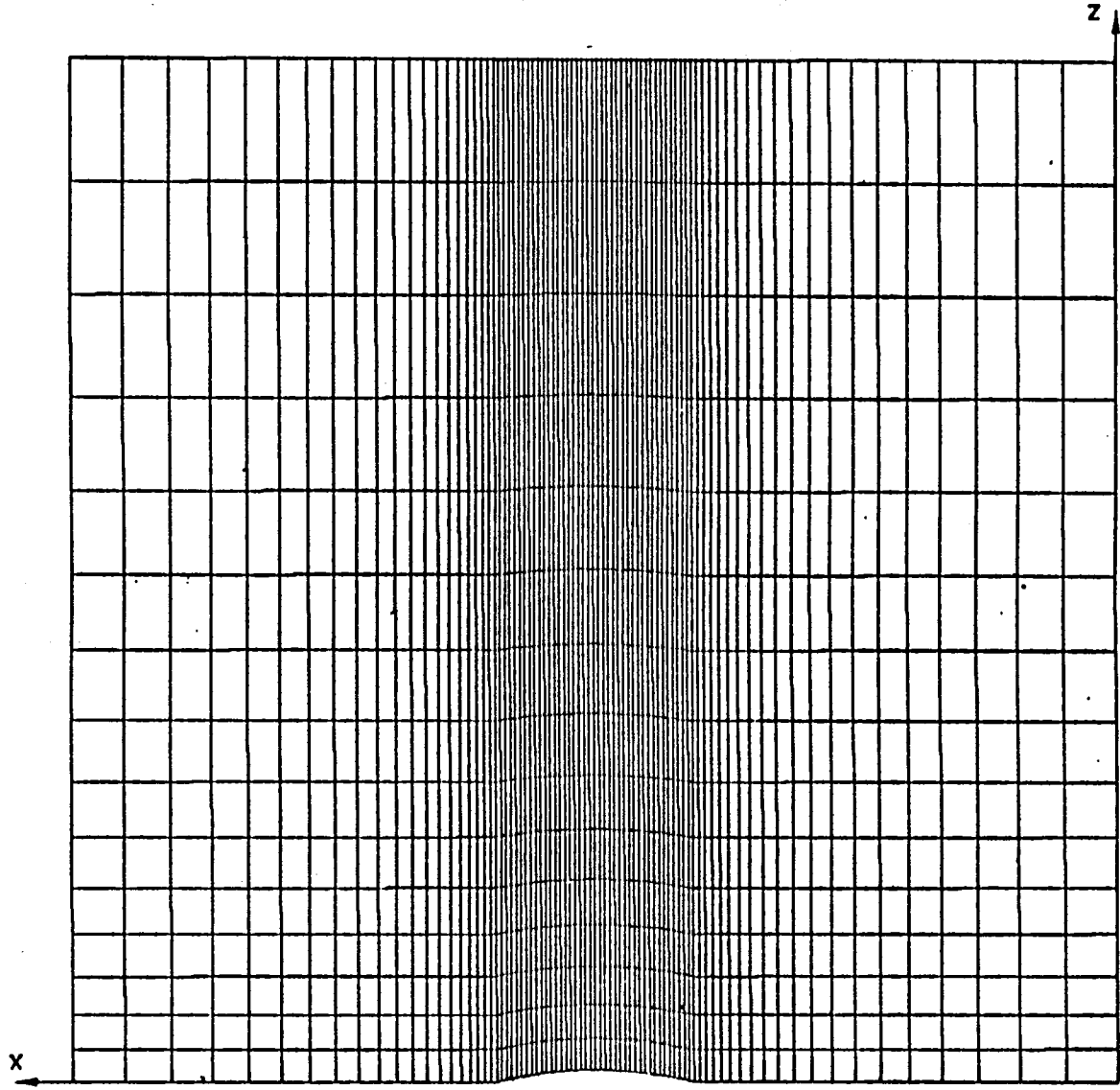


Figure 30. Grid-F, .12 Parabolic Arc Airfoil,  $FX=1.125$ ,  $FZ=1.1$ .

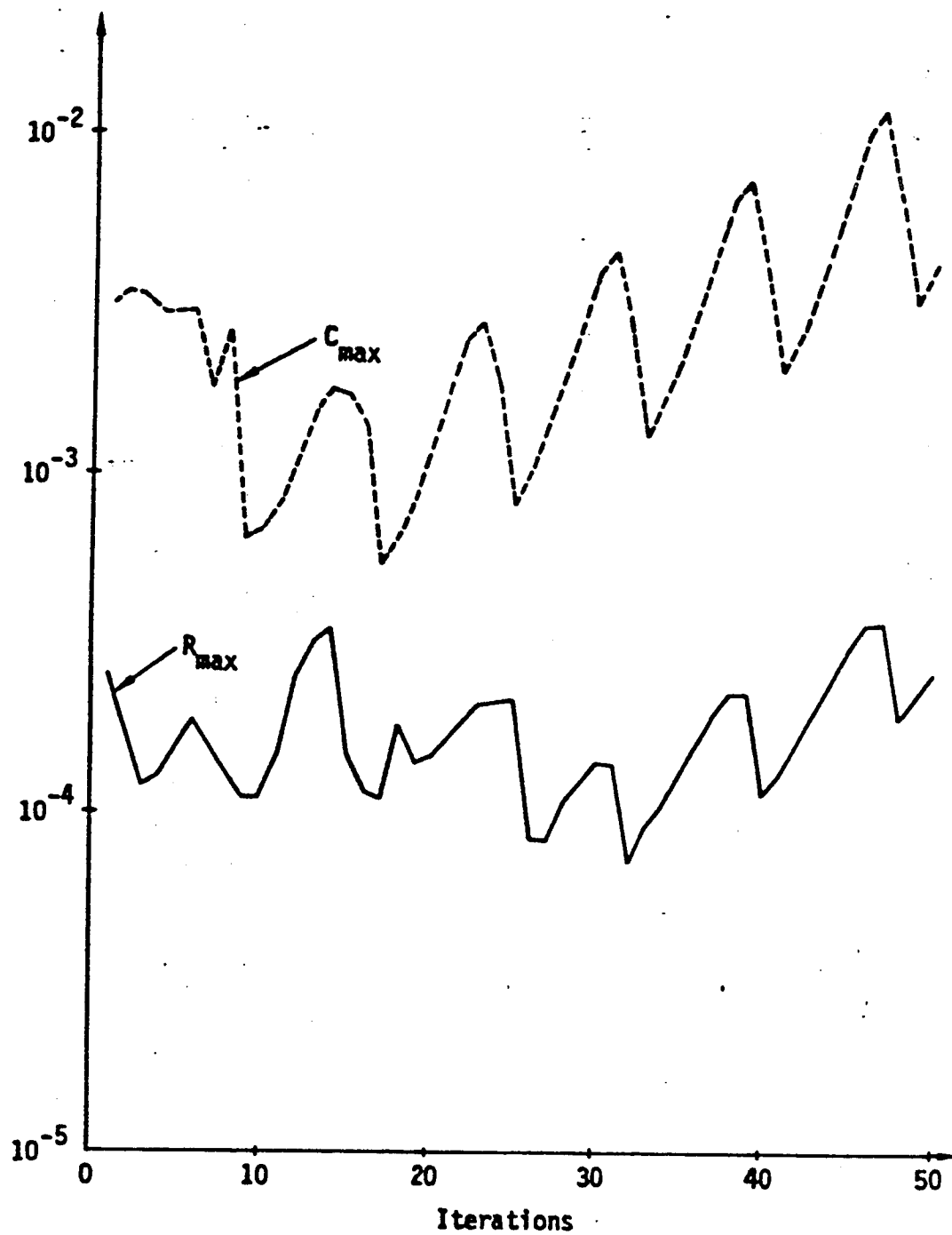


Figure 31. Growth of  $R_{max}$  and  $C_{max}$  for Flow Over a .12 Parabolic Arc Airfoil, Grid-F,  $M_\infty = .8$ .

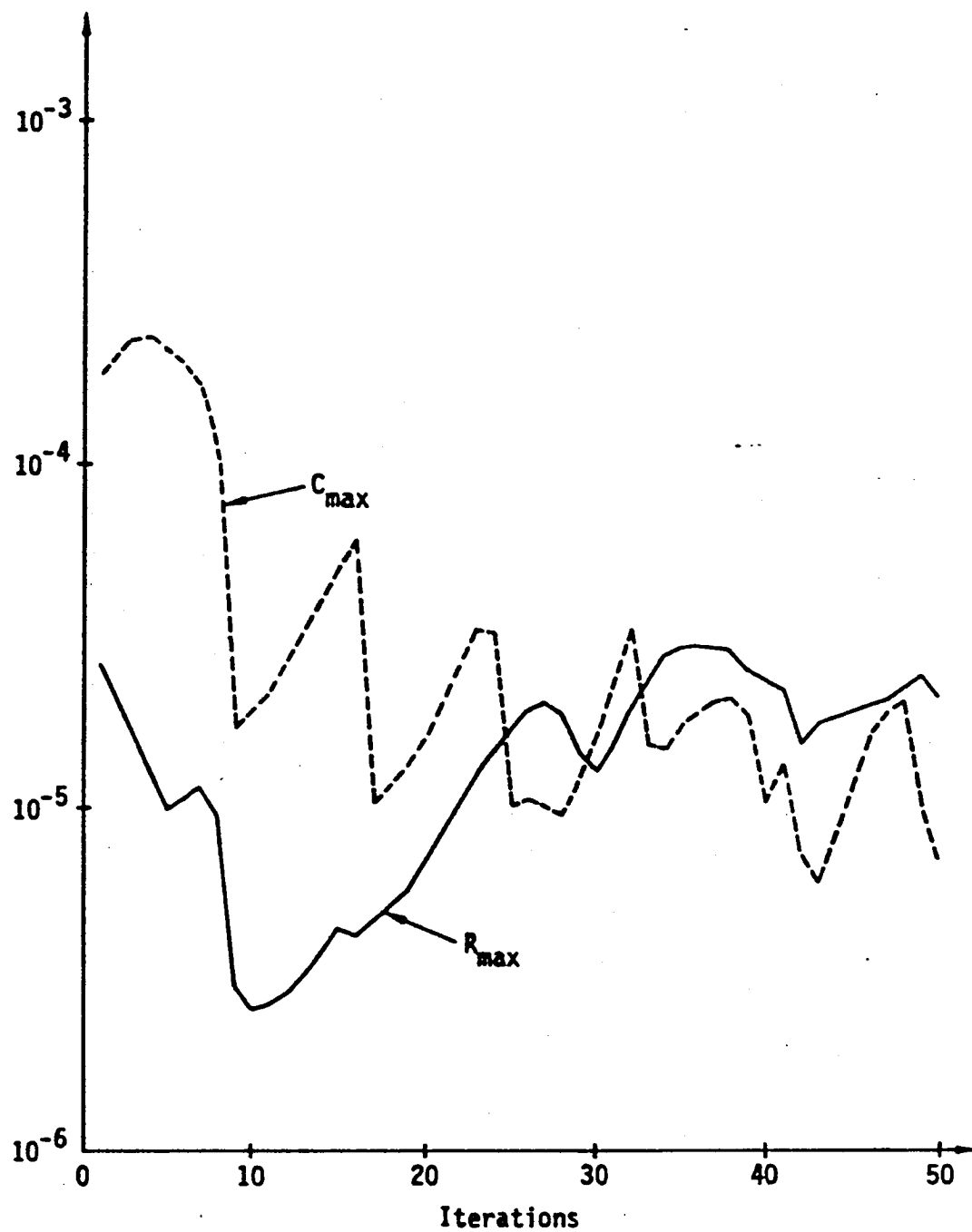


Figure 32. Growth of  $R_{\max}$  and  $C_{\max}$  for Flow Over a .12 Parabolic Arc Airfoil, Grid-F.  $M_{\infty}=.8$ ,  $\omega=1$ .

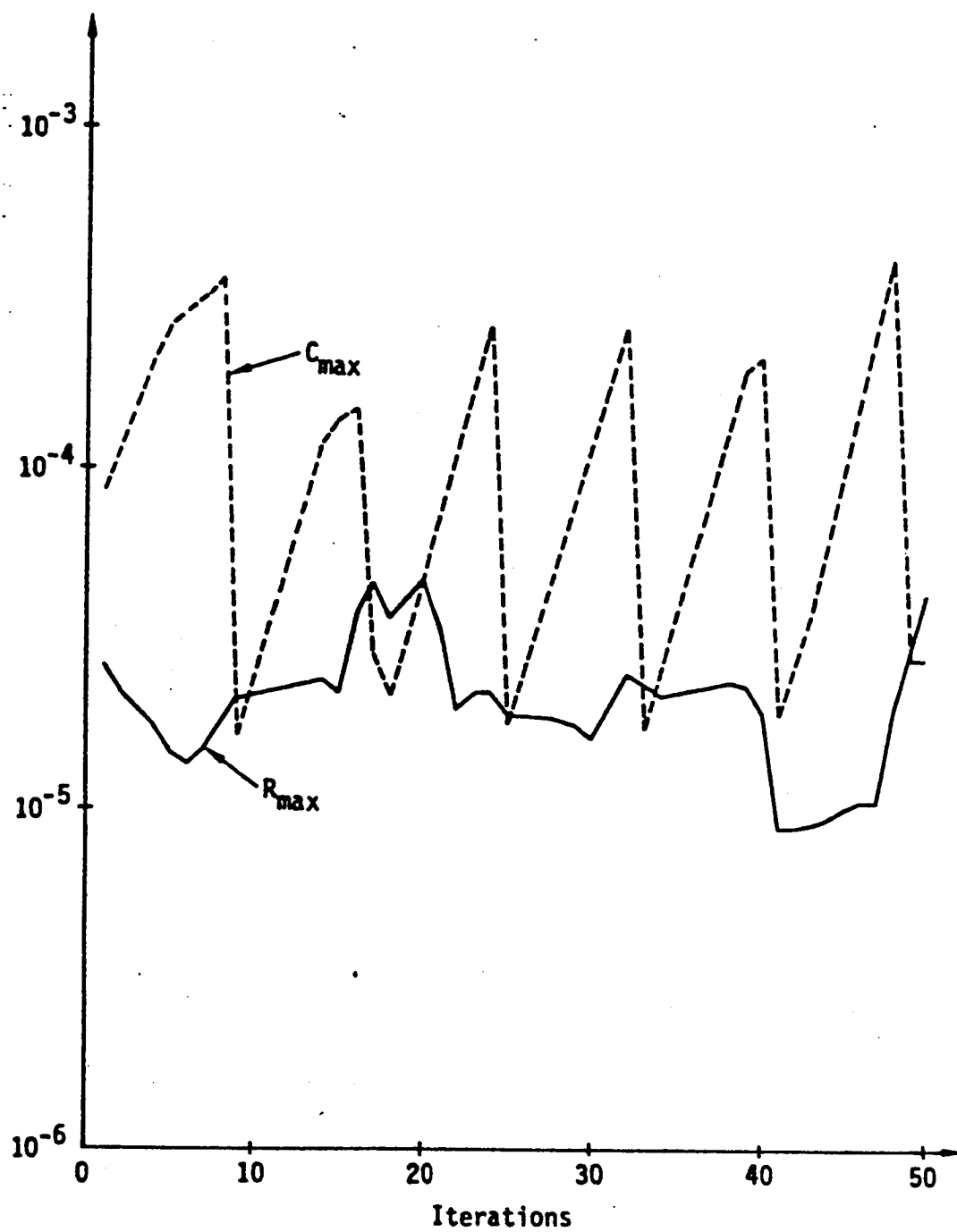


Figure 33. Growth of  $R_{\max}$  and  $C_{\max}$  for Flow Over a .12 Parabolic Arc Airfoil, Grid-F,  $M_{\infty}=.8$ ,  $\alpha_L=1$ ,  $\alpha_H=20$ .

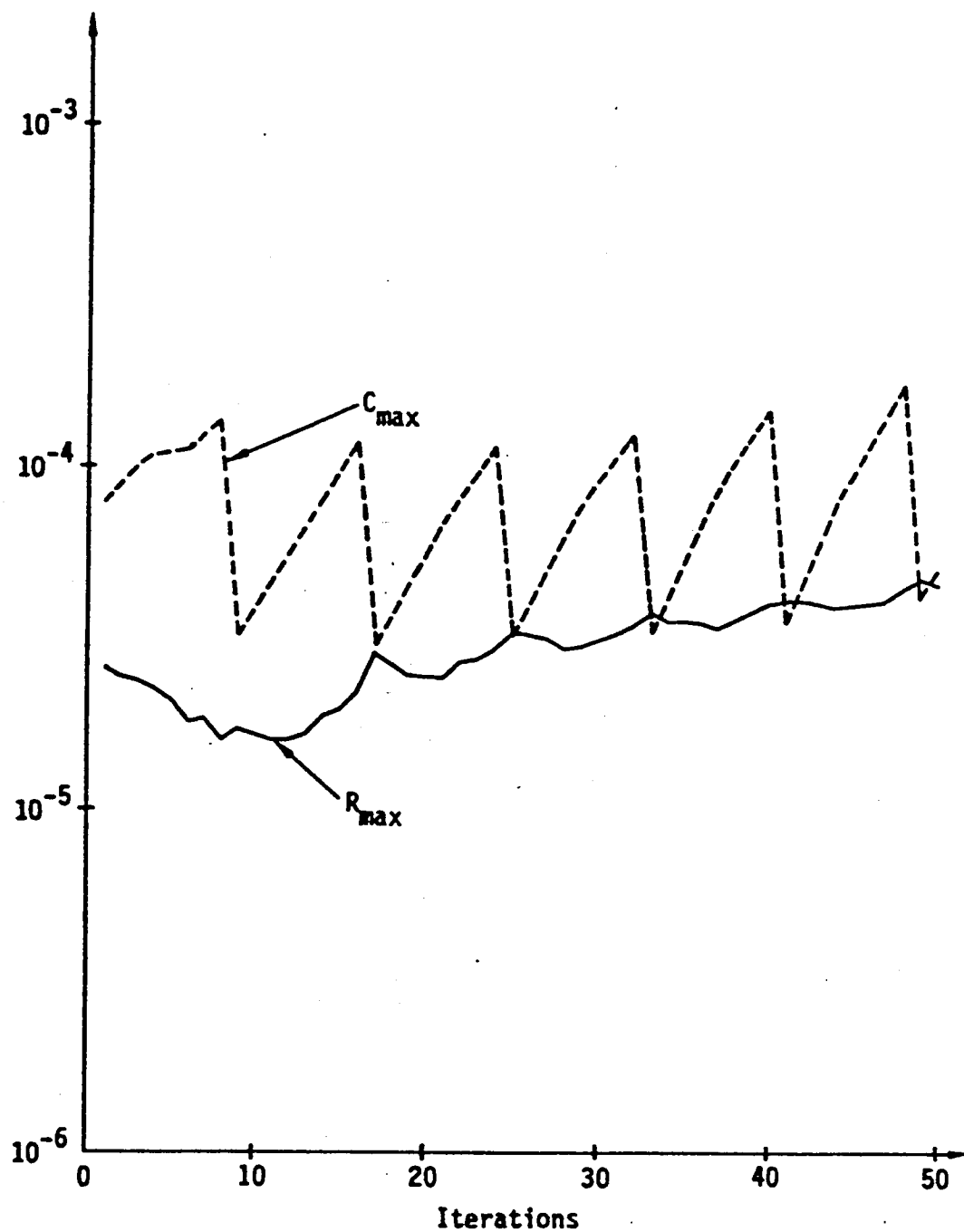


Figure 34. Growth of  $R_{max}$  and  $C_{max}$  for Flow Over a .12 Parabolic Arc Airfoil, Grid-F,  $M_{\infty}=.8$ ,  $\beta_{\xi}=1$  Everywhere.

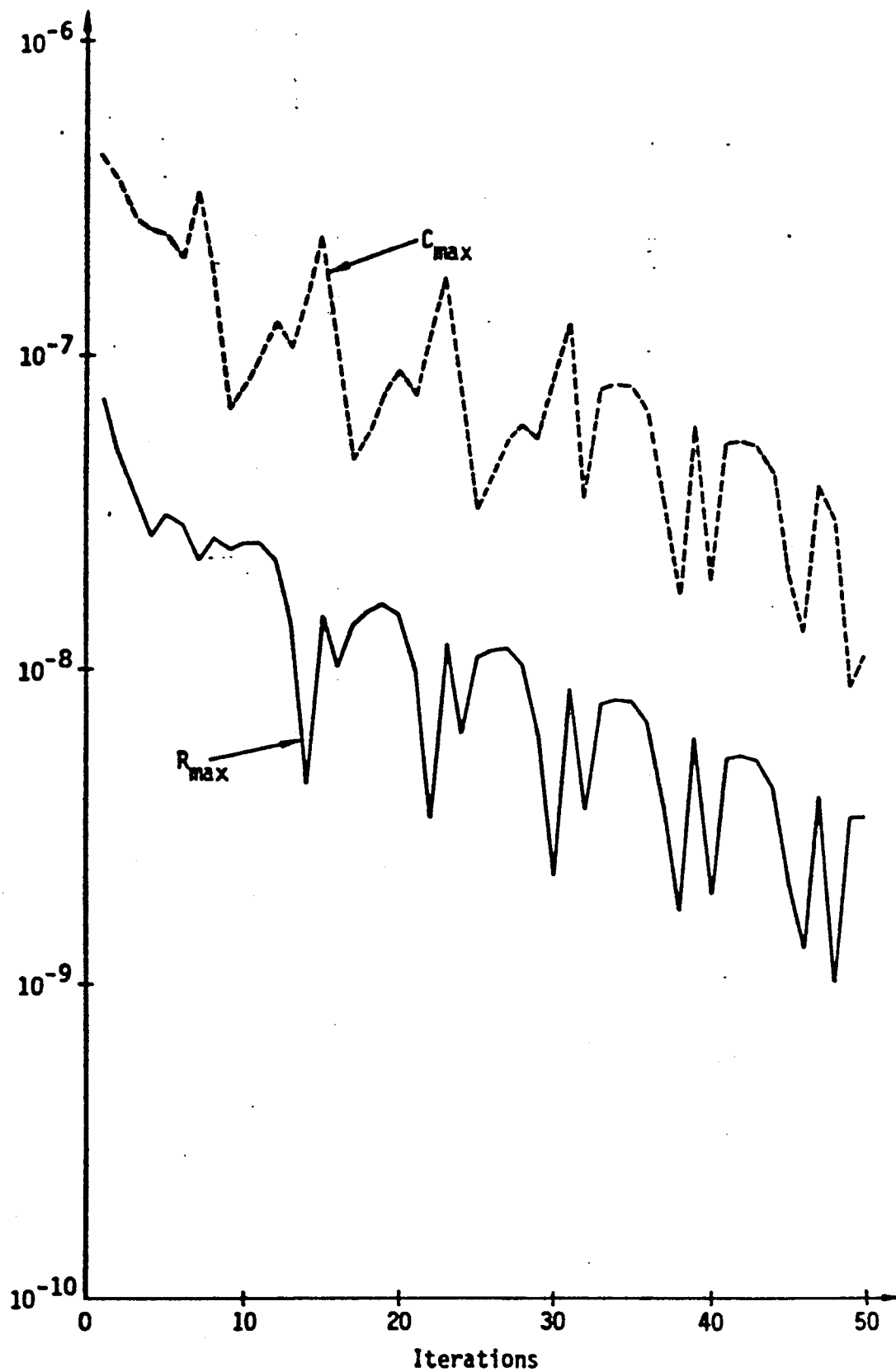


Figure 35. Convergence Histories for Flow Over a .12 Parabolic Arc Airfoil, Grid-F,  $M_{\infty} = .0001$ .



the solution is converging. Apparently the transonic calculations are more sensitive to the error induced by the grid effect. To make the transonic case work, it is clear that the Grid-F needs to be improved. Examining the central block (from  $i = 21$  to  $i = 61$ ) to the grid in Grid-F, one sees that the grid aspect ratio near the far field ( $z = -1$ ) is quite large.

A method to reduce the grid aspect ratio near  $z = -1$  is to reduce  $FZ$ , e.g.,  $FZ = 1$  so as to make the grid spacing in  $z$  uniform. In so doing, the grid aspect ratio near the airfoil surface is getting worse (Grid-G, Figure 36). This grid is not good enough to make the solution converge for  $M_\infty = .8$  (Figure 37). A better method to reduce the grid aspect ratio at  $z = -1$  is to redistribute the grid points along  $z = -1$  (e.g., uniform grid spacing in  $x$  along  $z = -1$ , grid-H, Figure 38).

For this grid, we also readjusted the block boundaries slightly so that the airfoil leading edge is at  $x = .4$ , and trailing edge at  $x = .6$ . The convergence histories in Figure 39 shows that the  $R_{\max}$  is reduced by three order of magnitude in 119 iterations. In Figure 39,  $R_{\max}$  and  $C_{\max}$  are plotted for every 8 iterations, corresponding to  $\alpha = \alpha_L$  in the eight-element sequence. Also shown in Figure 39 is the convergence history for the number of supersonic points NSUP. The NSUP tends to be overshoot within the first eight  $\alpha$  elements of the iterations, but it converges to its final value in only 27 iterations. This shows that the size of the supersonic pocket is stabilized very fast. This is typical for AF2 scheme in transonic flow calculations

The surface  $C_p$  distribution is given in Figure 40 and compared with the solutions using TCAS and LTRAN2. LTRAN2 used a small disturbance formulation. This is the main cause of the deviation of the LTRAN2 solution from the other two solutions. General agreement is quite good and the magnitude of the small difference between the solutions of TCAS and TWINGB is the same as for the subcritical case.

## B. Test Cases For Wing Problems

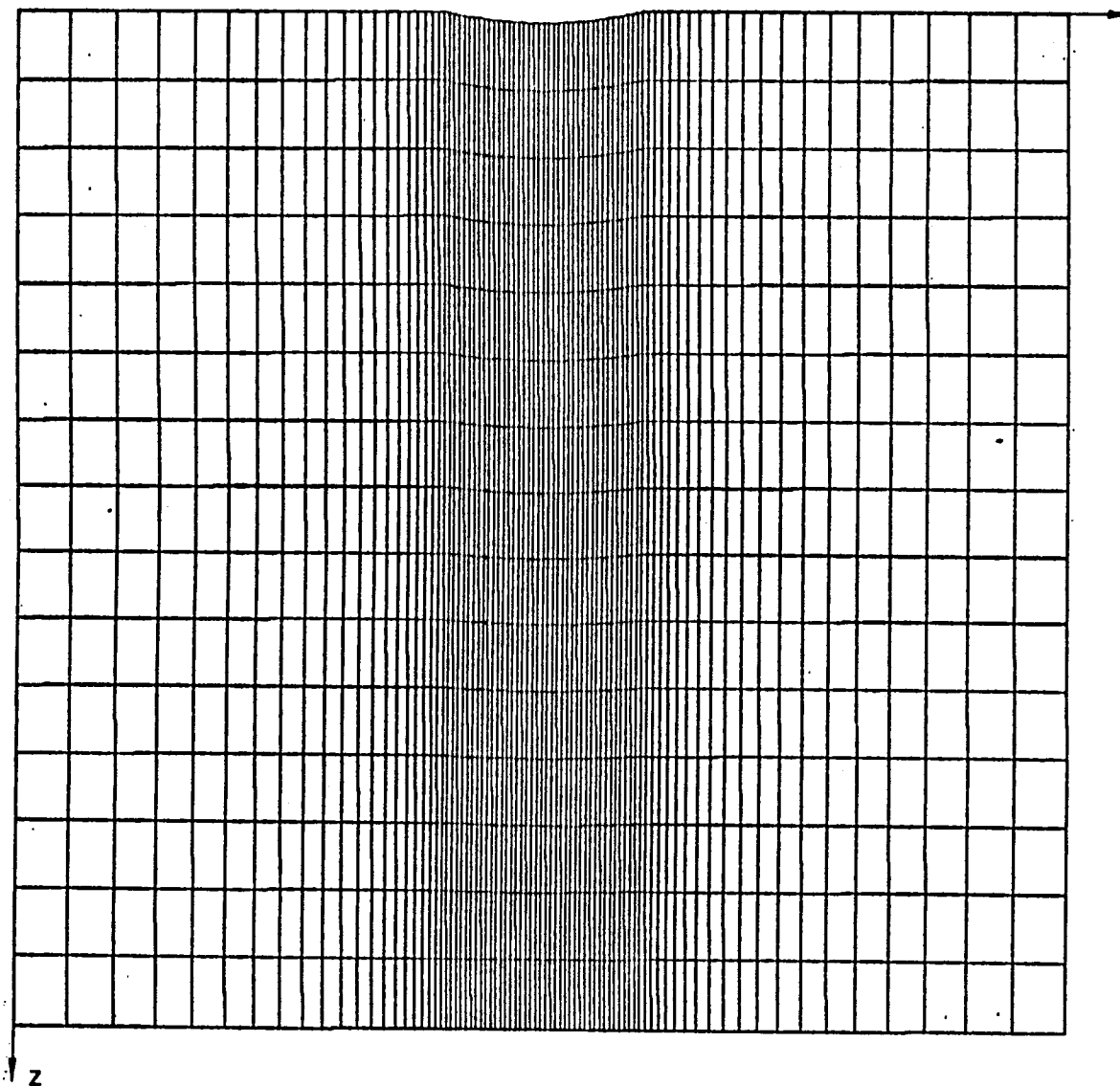


Figure 36. Grid-G, .12 Parabolic Arc Airfoil,  $FX=1.125$ ,  $FZ=1$ .

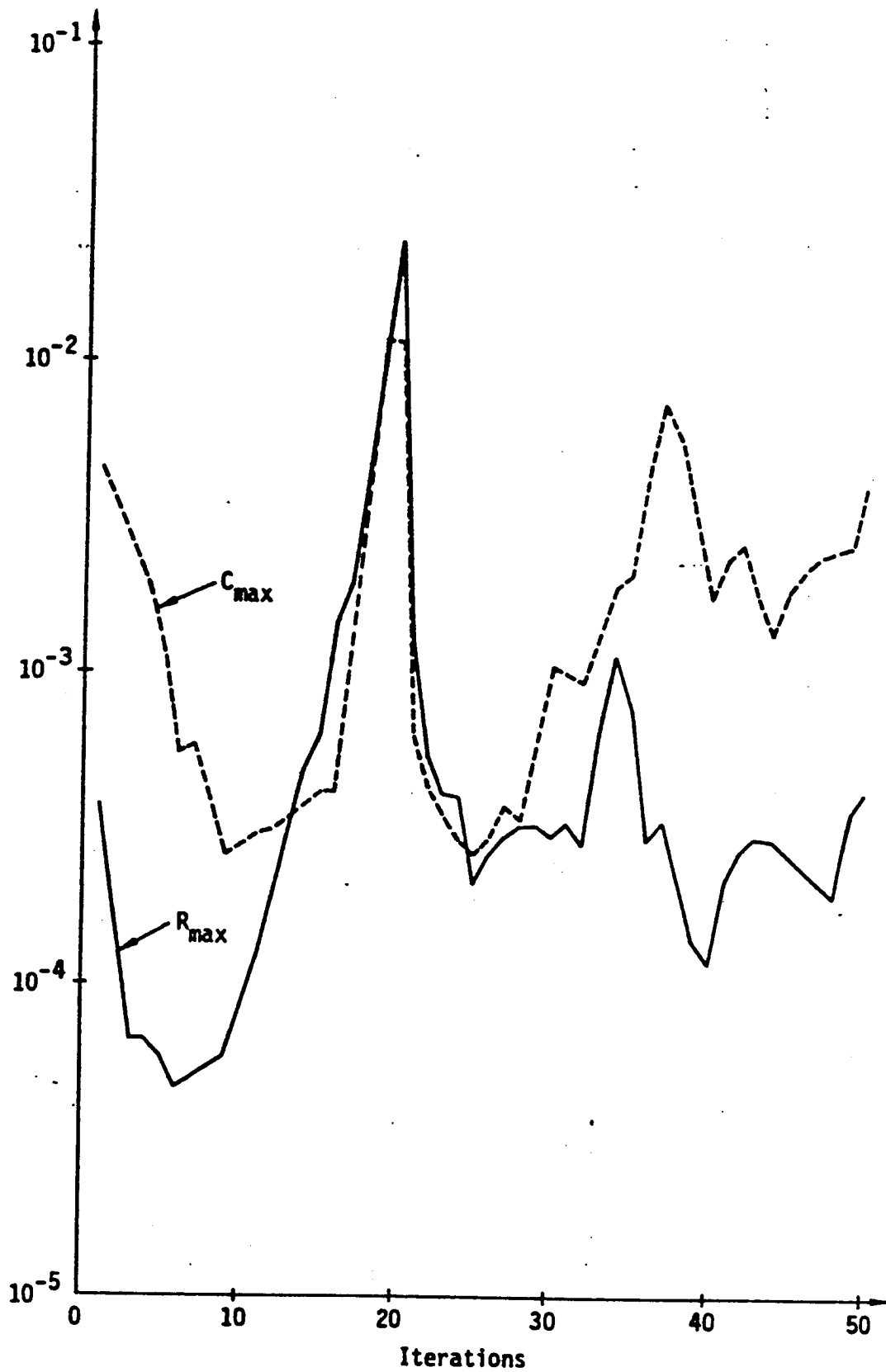


Figure 37. Growth of  $R_{\max}$  and  $C_{\max}$  for Flow Over a .12 Parabolic Arc Airfoil, Grid-G,  $M_{\infty}=.8$ .

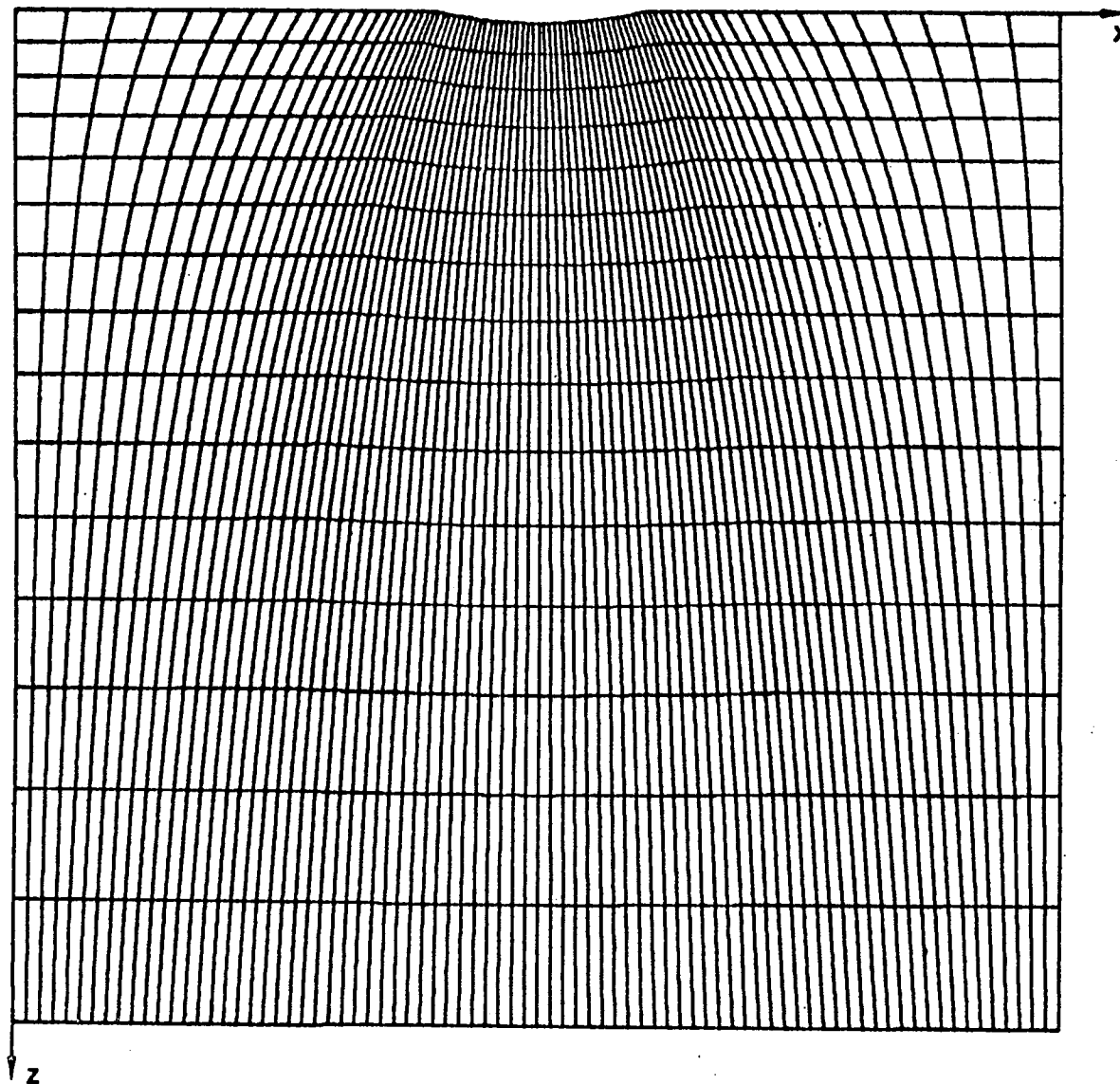


Figure 38. Grid-H, .12 Parabolic Arc Airfoil,  $FX1=1.129$ ,  $FX2=1$ ,  $FZ=1.1$ .

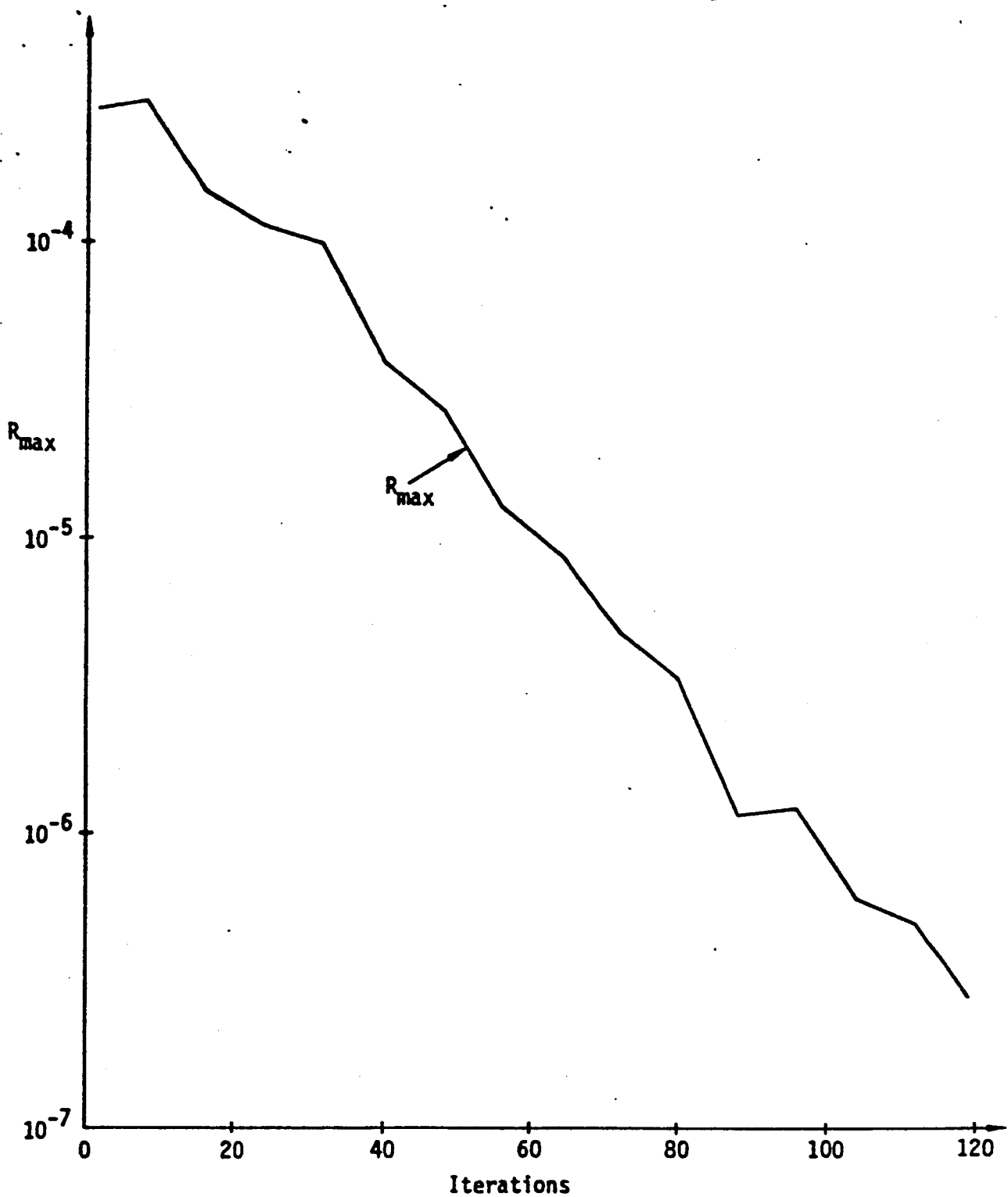


Figure 39a. Convergence History of  $R_{\max}$  for Flow Over a .12 Parabolic Arc Airfoil, Grid-H,  $M_{\infty}=8$ .

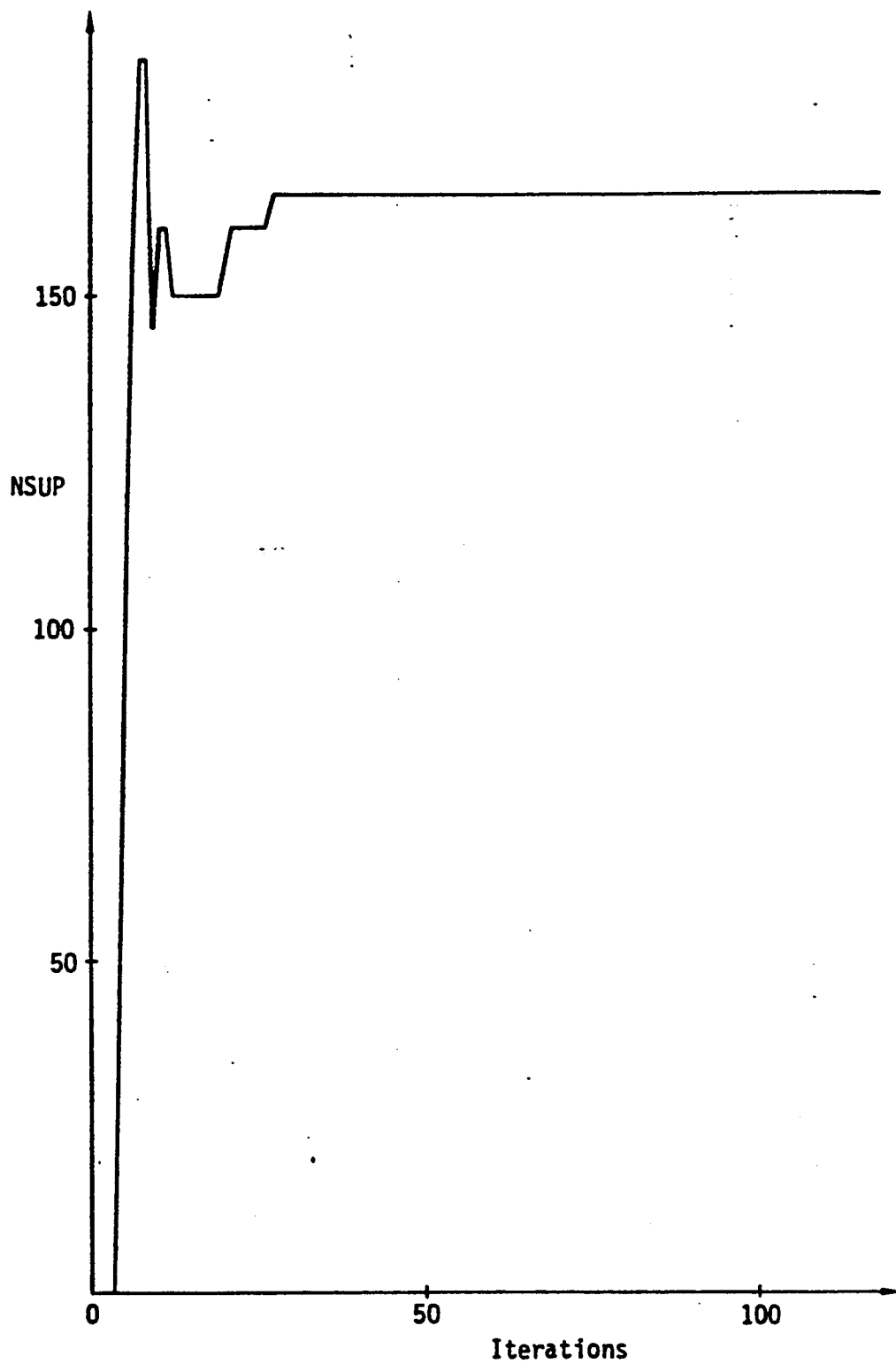


Figure 39b. Convergence History of NSUP for Flow Over a .12 Parabolic Arc Airfoil, Grid-H,  $M_\infty = .8$ .

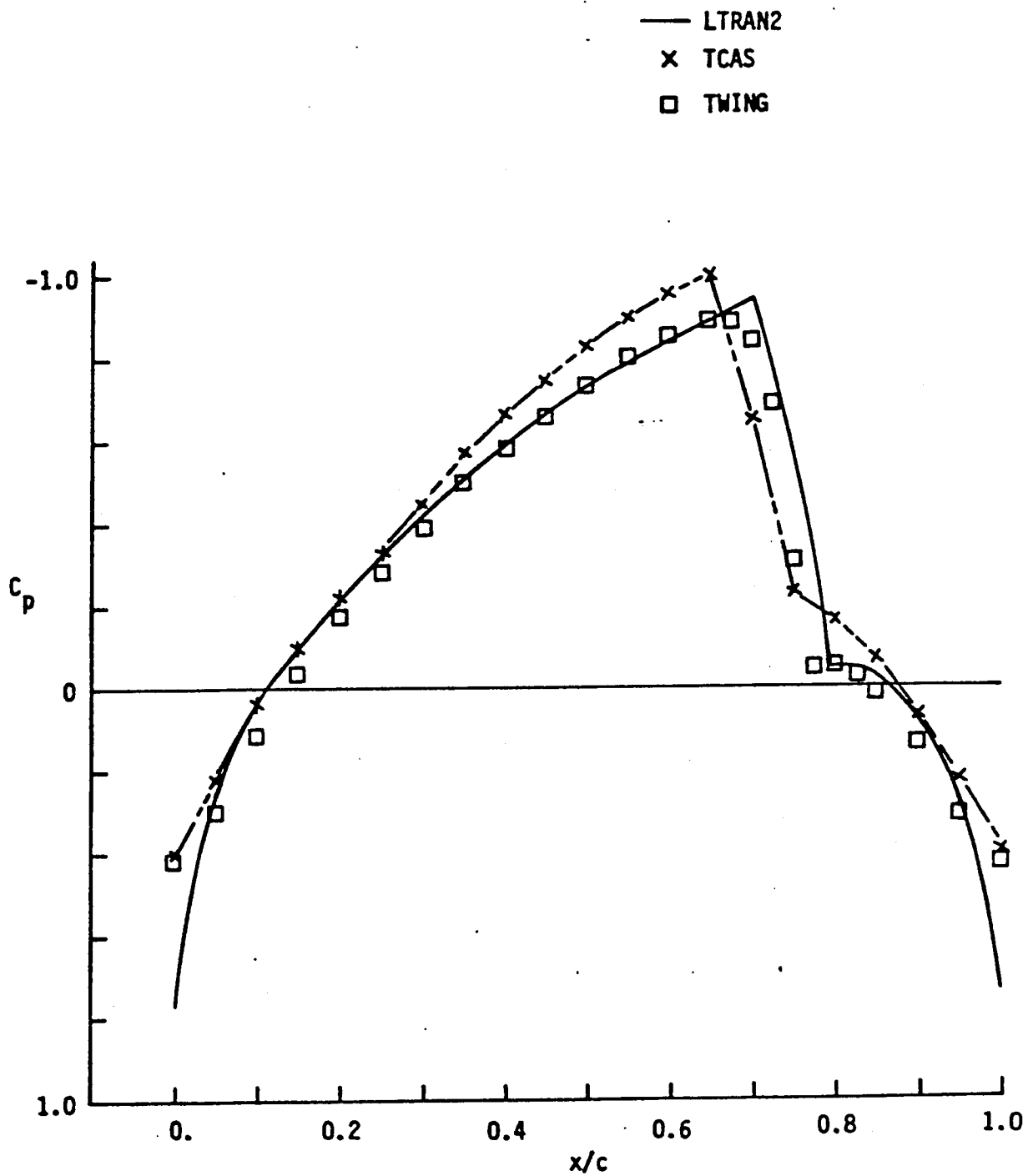


Figure 40. Surface  $C_p$  Distributions for Flow Over a .12 Parabolic Arc Airfoil, Grid-H,  $M_\infty = .8$ .

1. Uniform Flow Over a Swept Flat Plate.

This case was not tested in view of the success of the cases of flow over parabolic arc airfoils.

2. A Swept Wing With a .03 Parabolic Arc Wing Section.

A high aspect ratio ( $AR = 9.5$ ) wing has been studied first. This wing is non-tapered, has a 30 degree sweep and is mounted between two parallel walls extended to infinity. The planform of this wing is identified as planform-A.

In each of the 20 spanwise stations, a grid similar to that in Figure 27 (Grid-E) is used. The block boundaries are adjusted slightly. For the wing root section, the wing leading edge is at  $x = .4$  and trailing edge at  $x = .6$ .

This wing is analyzed at  $M_\infty = .75$ . The convergence histories given in Figure 41 shows that it takes 103 iterations to reduce the  $R_{max}$  by three orders of magnitude. The 3D effects slow down the convergence and the resulting convergence speed is two and a half times slower than the 2D counterpart. This is to be expected since the variation of the flow in the spanwise direction may induce further error due to the approximations employed in TWINGB. Since these errors are grid dependent, the convergence should be improved when using a better grid. An improved grid (compared with Grid-E) is displayed in Figure 42 as Grid-I. Using Grid-I for flow calculations, the convergence histories given in Figure 43 shows that  $R_{max}$  reduced by three orders of magnitude in only 54 iterations. The convergence speed is improved by a factor greater than two due to grid improvement. The surface  $C_p$  results are given in Figure 44 for the wing root, wing mid-span, and wing tip stations.

The low aspect ratio ( $AR = 1.9$ ) case has also been analyzed at  $M_\infty = .75$  and this case is expected to produce a more pronounced 3D effect. The planform of this wing is called planform-B which has a 30 degree sweep. The wing is non-tapered and it is mounted between two parallel walls extended to infinity. In each of the 20 spanwise stations, Grid-I (Figure 42) is used.



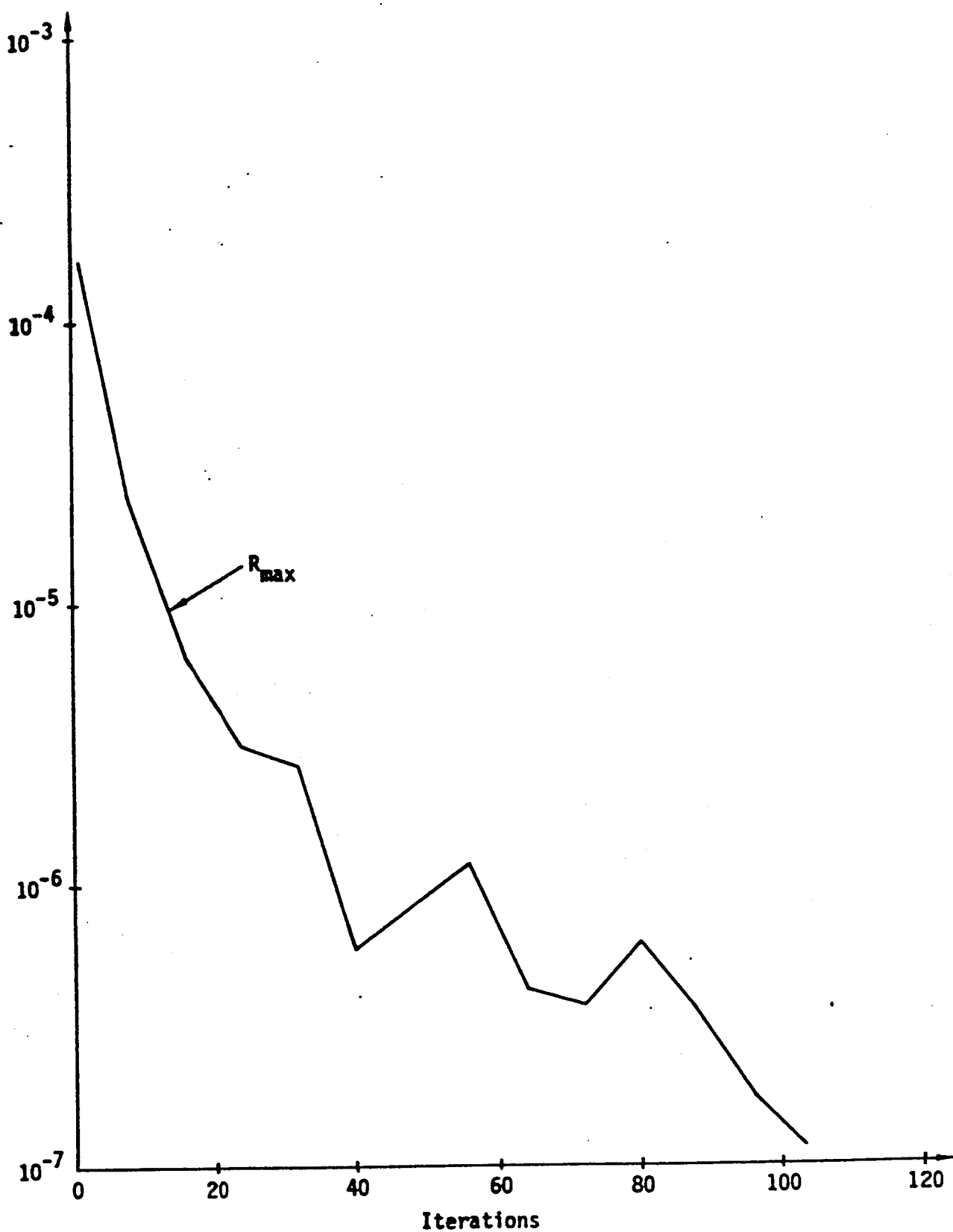


Figure 41. Convergence History for Flow Over a Swept Wing With a .03 Parabolic Arc Wing Section, Planform-A, Grid-E,  $M_{\infty}=.75$ .

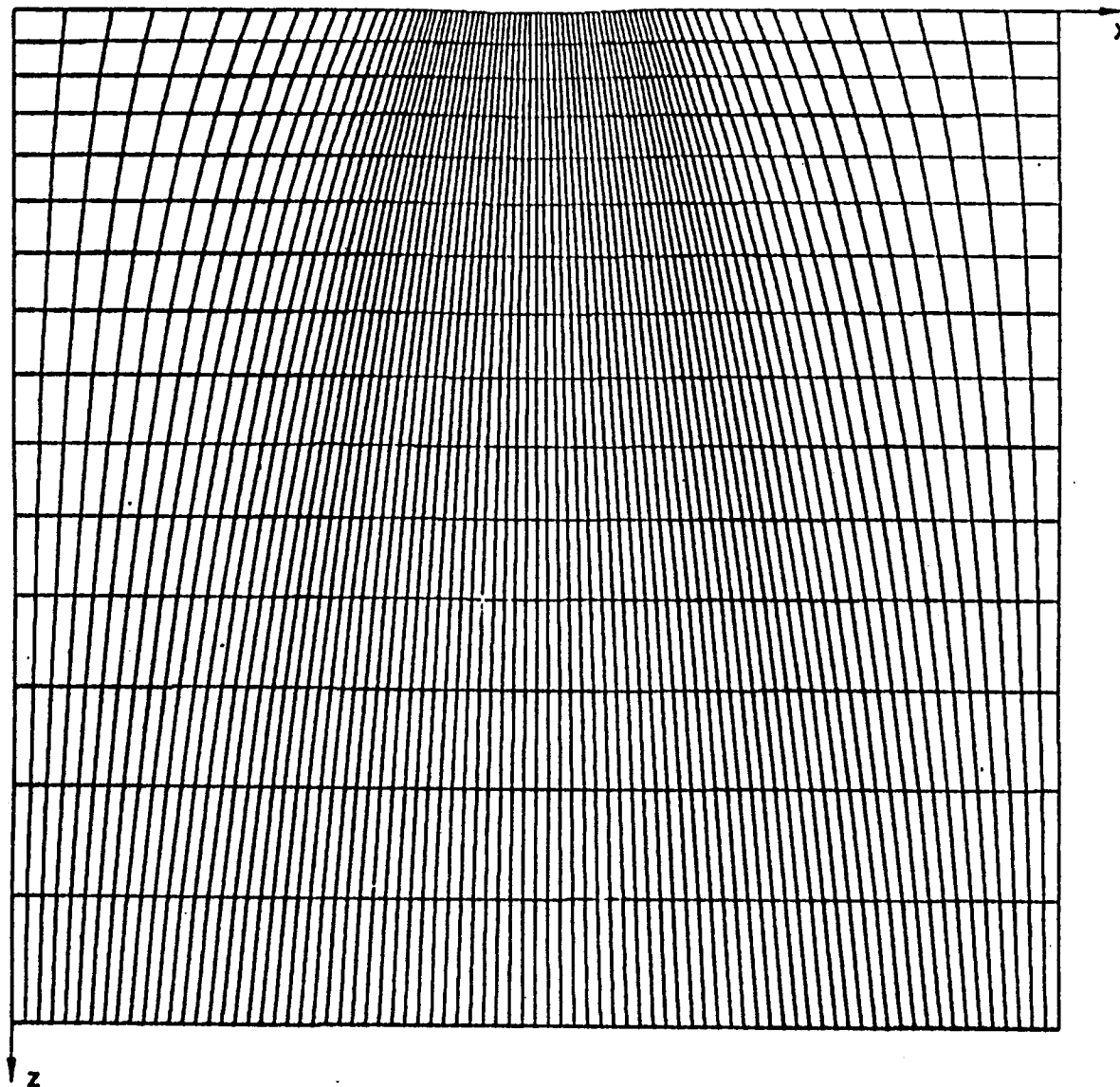


Figure 42. Grid-I, .03 Parabolic Arc Wing Section,  $FX1=1.129$ ,  $FX2=1$ ,  $FZ=1.1$ .

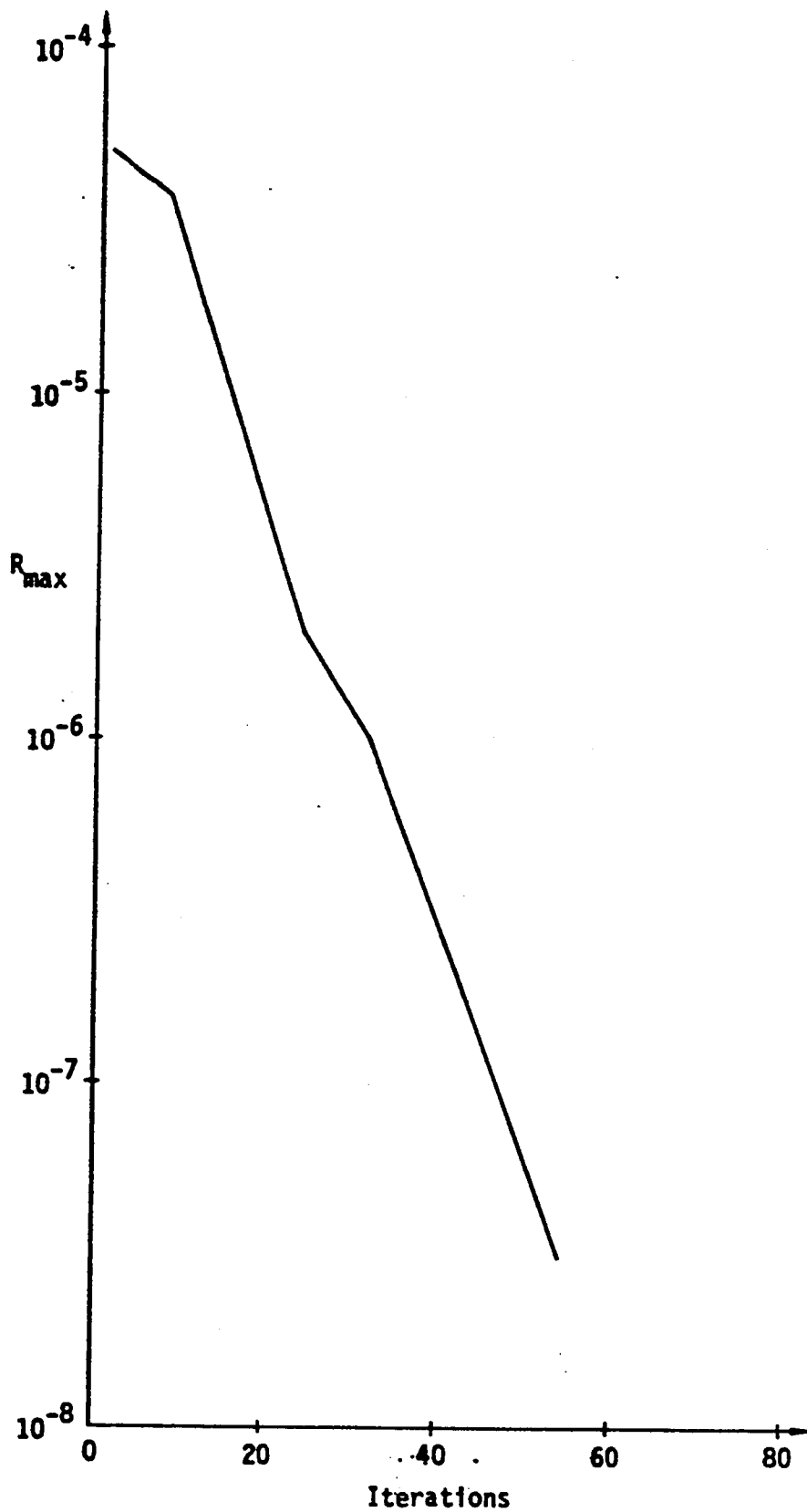


Figure 43. Convergence History for Flow Over a Swept Wing With a .03 Parabolic Arc Wing Section, Planform-A, Grid-I,  $M_{\infty}=.75$ ,  $Ar=9.5$ .

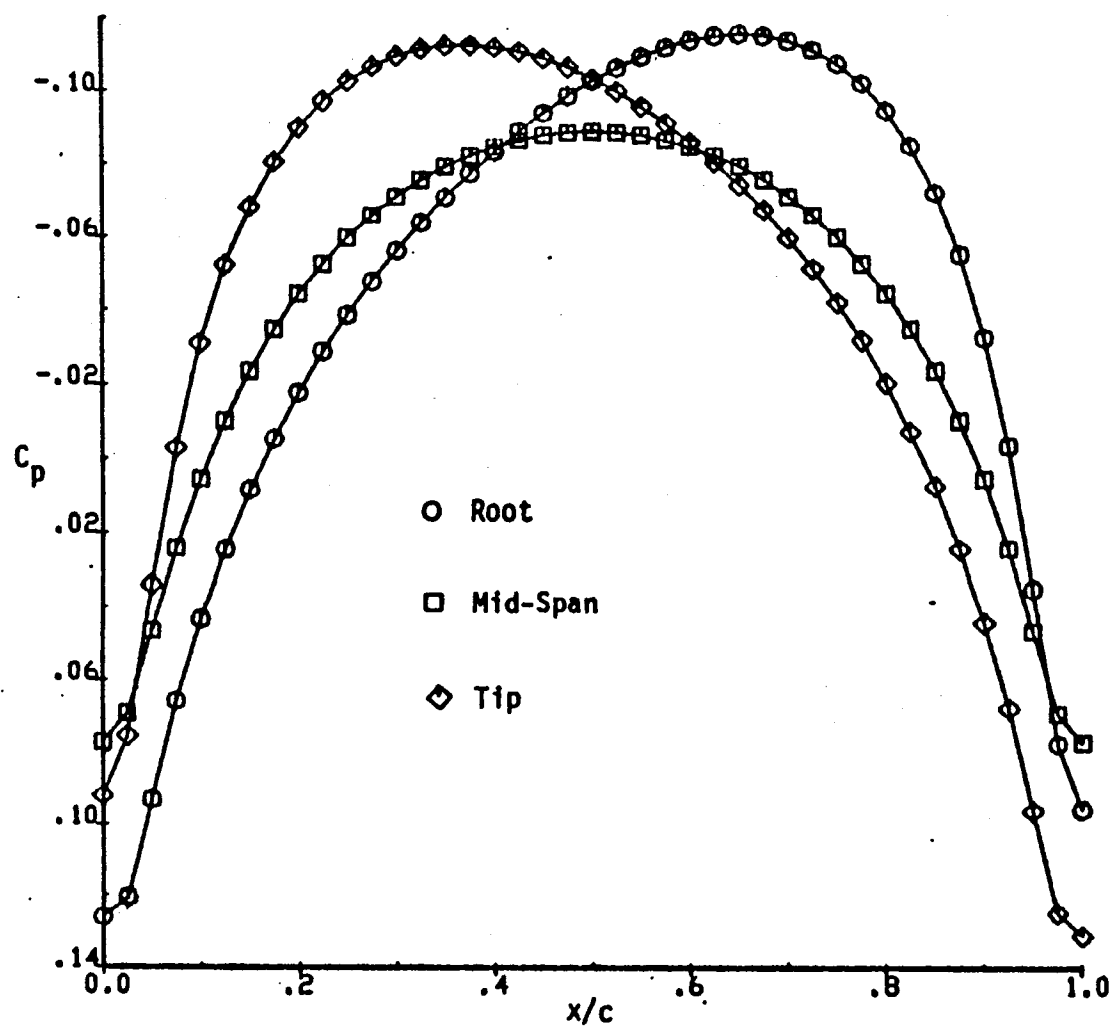


Figure 44. Surface  $C_p$  Distributions for Flow Over a Swept Wing With a .03 Parabolic Arc Wing Section, Planform-A, Grid-I,  $M_\infty=.75$ ,  $AR=9.5$ .

The numerical convergence for this case is very slow (274 iterations, see Figure 45). The pronounced 3D effects affect the convergence and the resulting convergence speed is two and a half times slower than the high aspect ratio wing case. The surface  $C_p$  distributions are given in Figure 46 for the wing root, wing mid-span, and wing tip stations.

### 3. A Swept Wing with a 0.12 Thick Parabolic Arc Wing Section

For the high aspect ratio wing case, planform-A ( $AR = 9.5$ ) is used. In each of the 20 spanwise stations, Grid-H (Figure 38) is used.

The wing is analyzed at  $M_\infty = .8$ . The convergence histories given in Figure 47 show that it takes 123 iterations for the  $R_{max}$  to reduce by three order to magnitudes. However, the number of supersonic point NSUP converged in 53 iterations. This shows that the size of the supersonic pocket is established at very early stage of the iterations, similar to the 2D cases.

The shock wave position is plotted on the wing planform in Figure 48. As expected, the shock wave approaches both sidewalls nearly perpendicularly. Near center span, the shock is swept and is approximately parallel to the wing leading and trailing edges. The surface  $C_p$  results in the root, center span and tip planes are given in Figure 49.

For the low aspect ratio wing case, planform-B ( $AR = 1.9$ ) is used. Grid-H is used for each of the 20 spanwise stations. The wing is analyzed at  $M_\infty = .8$ .

Numerical convergence is extremely slow (Figure 50). The convergence of  $R_{max}$  by three orders of magnitude takes 373 iterations and the convergence of NSUP consumes 135 iterations. That is more than twice as slow as the high aspect ratio wing case. This is resulting from the error, induced from the approximation procedures in the flow solution algorithm, becoming more severe due to strong 3D effects.

The shock wave position is plotted on the wing planform in Figure 51. The shock wave approaches both sidewalls nearly perpendicularly. Near center span, the shock is swept and is approximately parallel to the wing leading

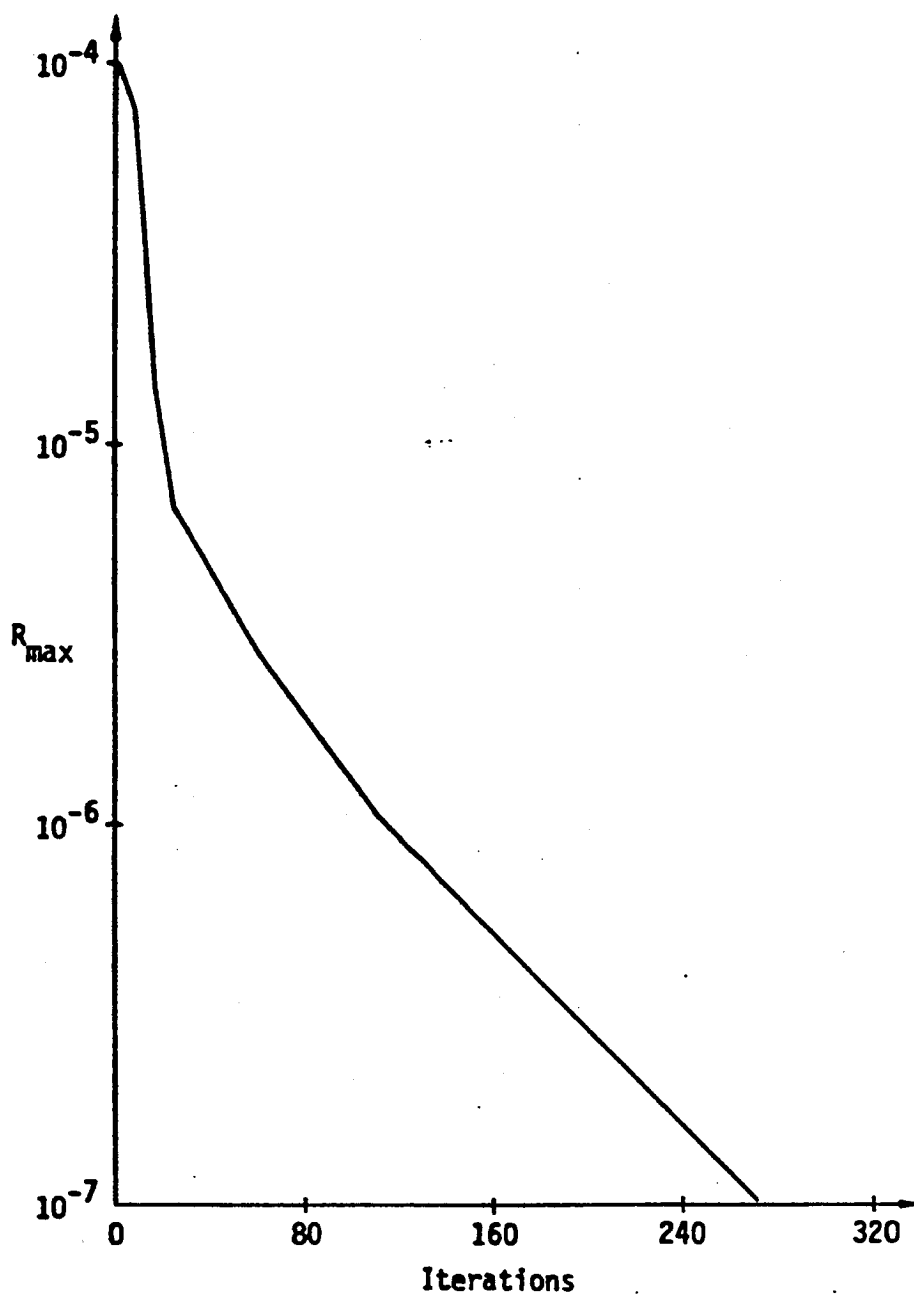


Figure 45. Convergence History for Flow Over a Swept Wing With a .03 Parabolic Arc Wing Section, Planform-B, Grid-I,  $M_{\infty}=.75$ ,  $AR=1.9$ .

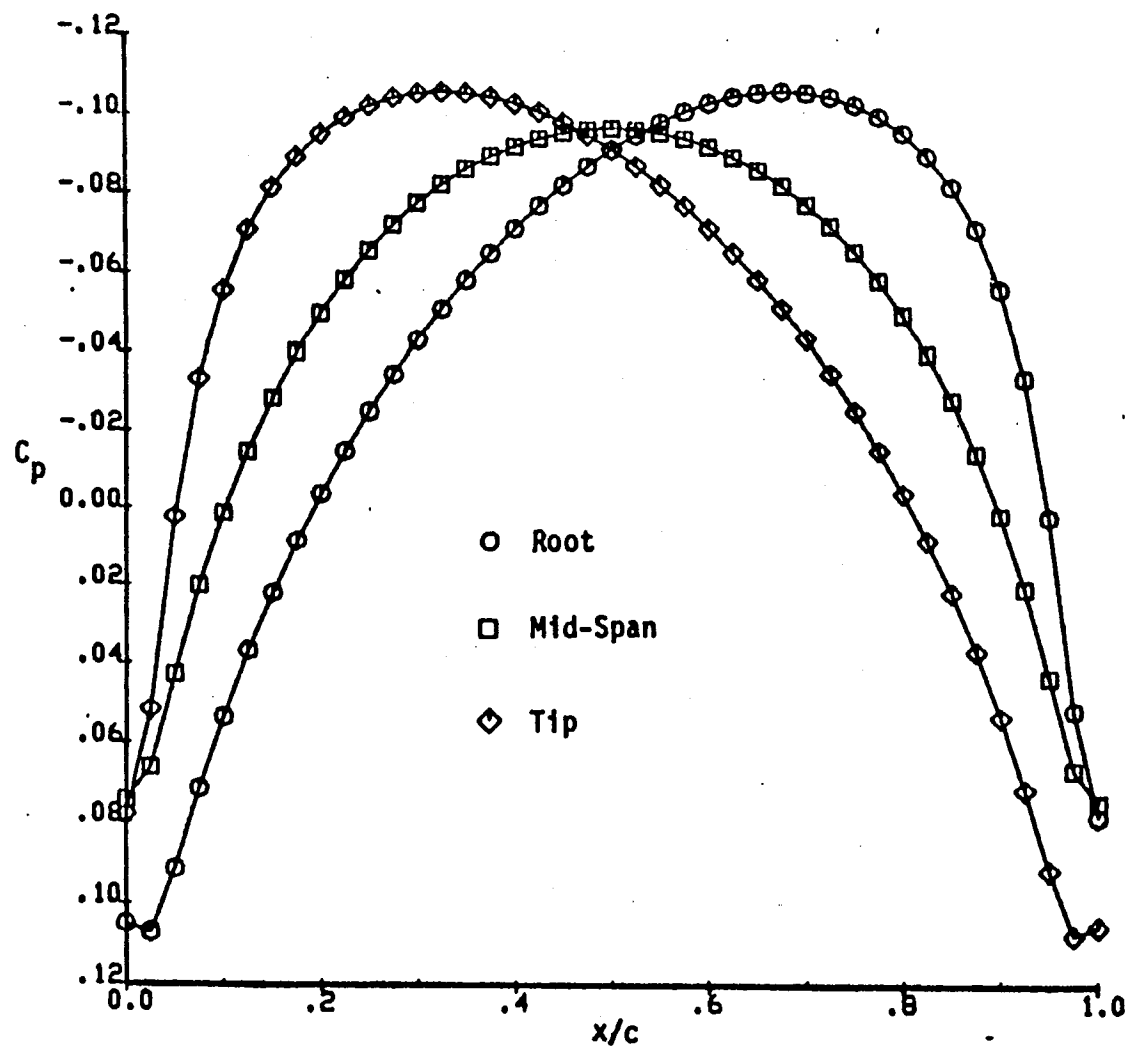


Figure 46. Surface  $C_p$  Distributions for Flow Over a Swept Wing With a .03 Parabolic Arc Wing Section, Planform-B, Grid-I,  $M_\infty = .75$ ,  $AR = 1.9$ .

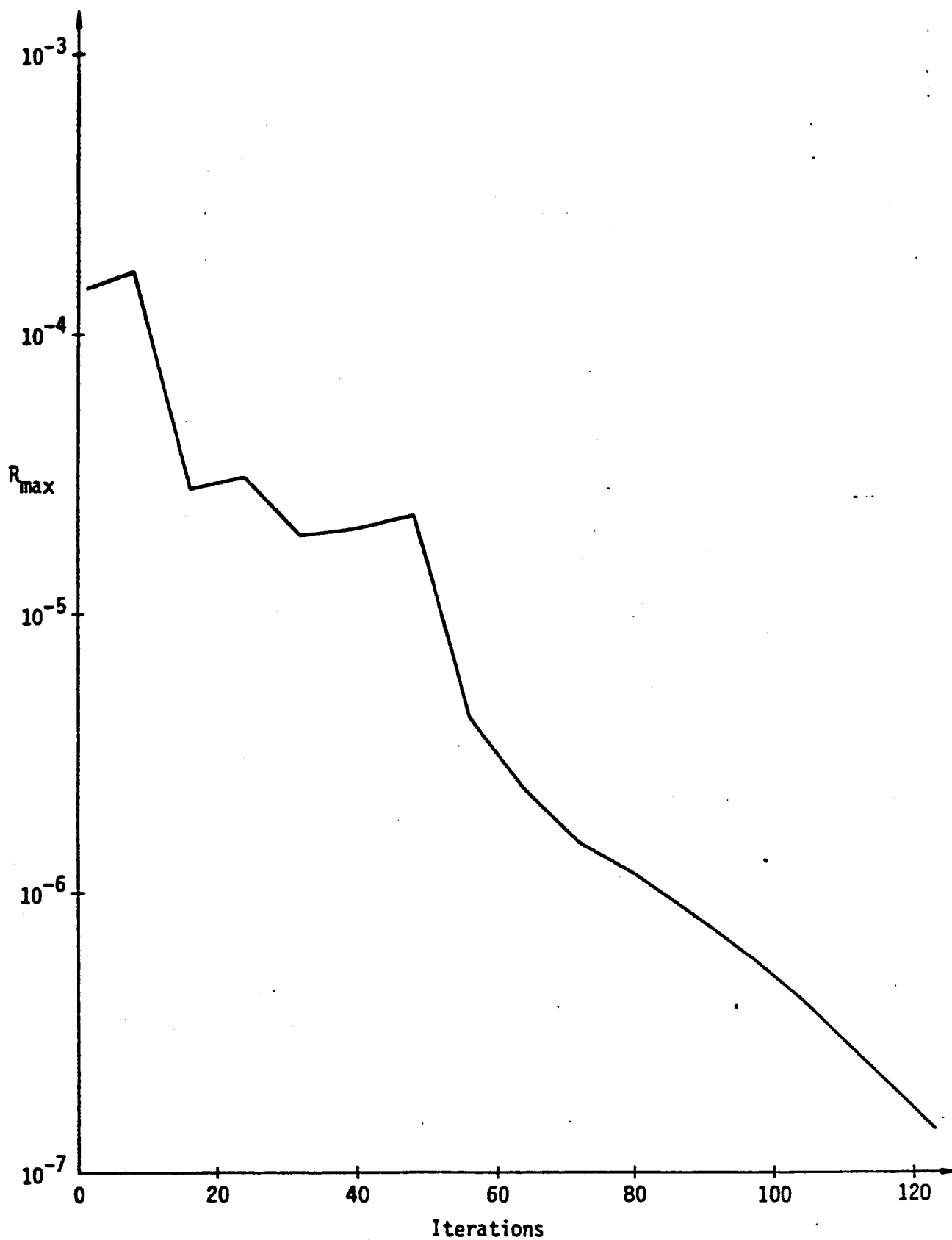


Figure 47a. Convergence History of  $R_{\max}$  for Flow Over a Swept Wing With a .12 Parabolic Arc Wing Section, Planform-A, Grid-H,  $M_{\infty}=.8$ ,  $AR=9.5$ .



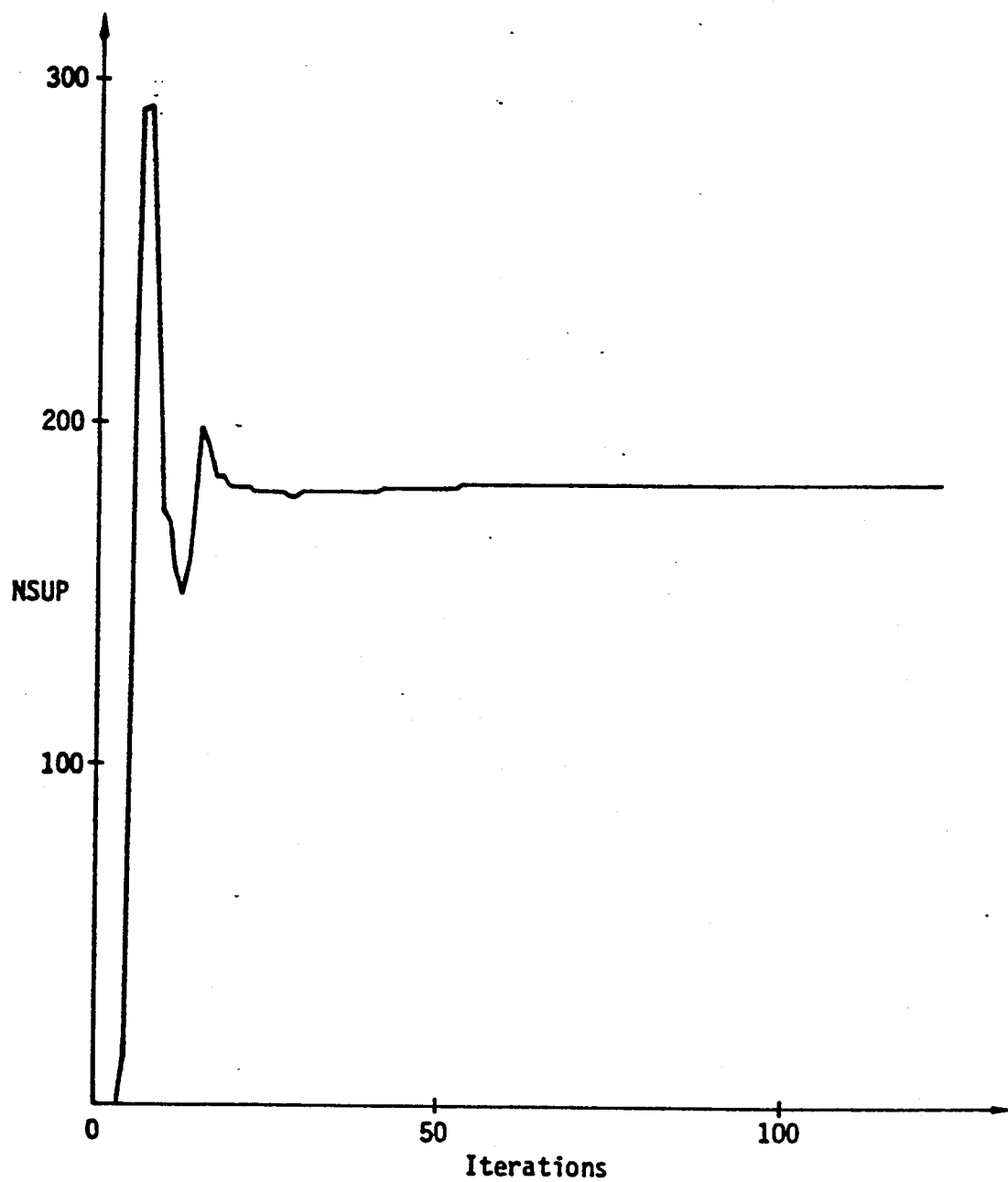


Figure 47b. Convergence History of NSUP for Flow Over a Swept Wing With a .12 Parabolic Arc Wing Section, Planform-A, Grid-H,  $M_\infty=.8$ ,  $AR=9.5$ .

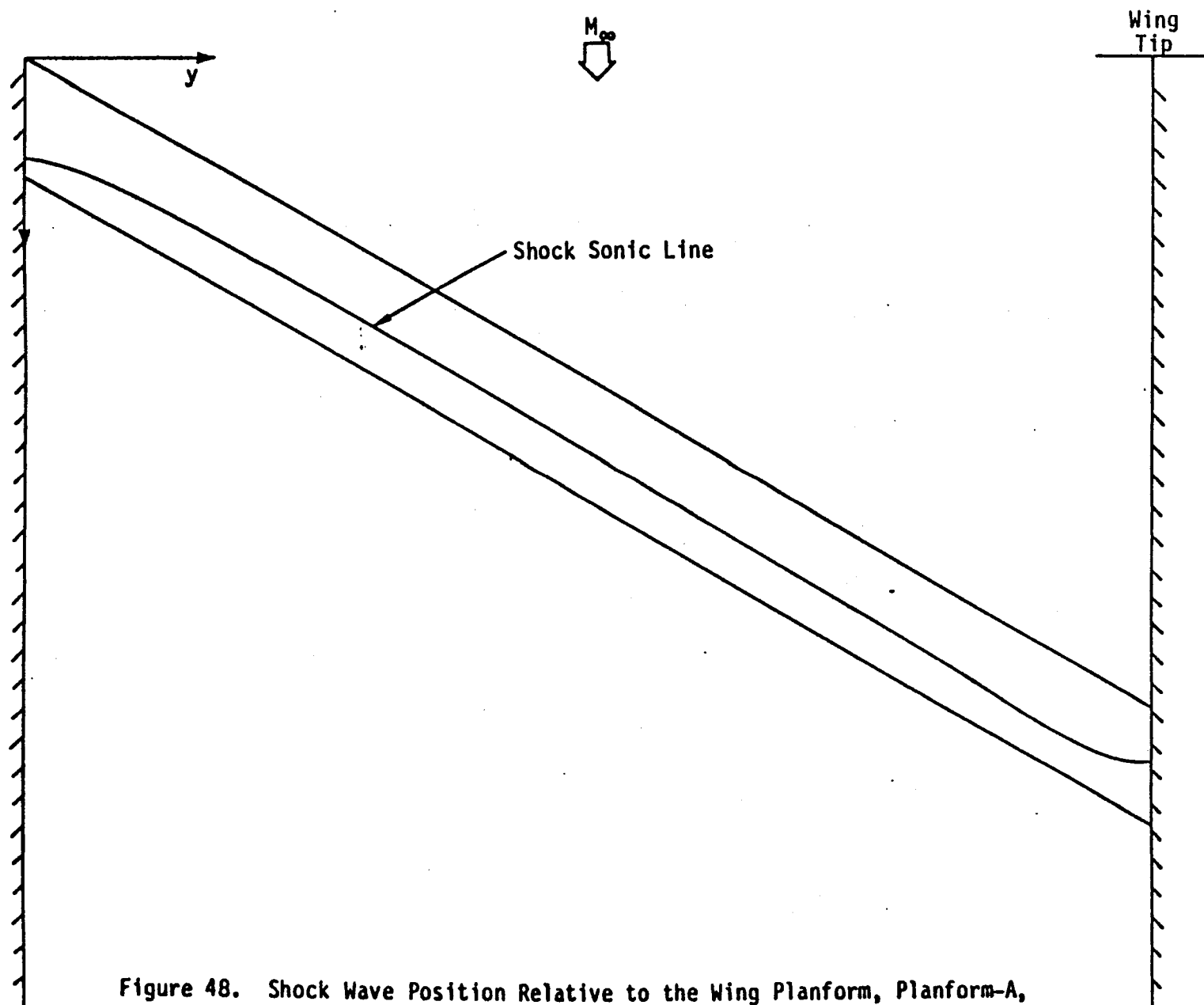


Figure 48. Shock Wave Position Relative to the Wing Planform, Planform-A, AR=9.5.

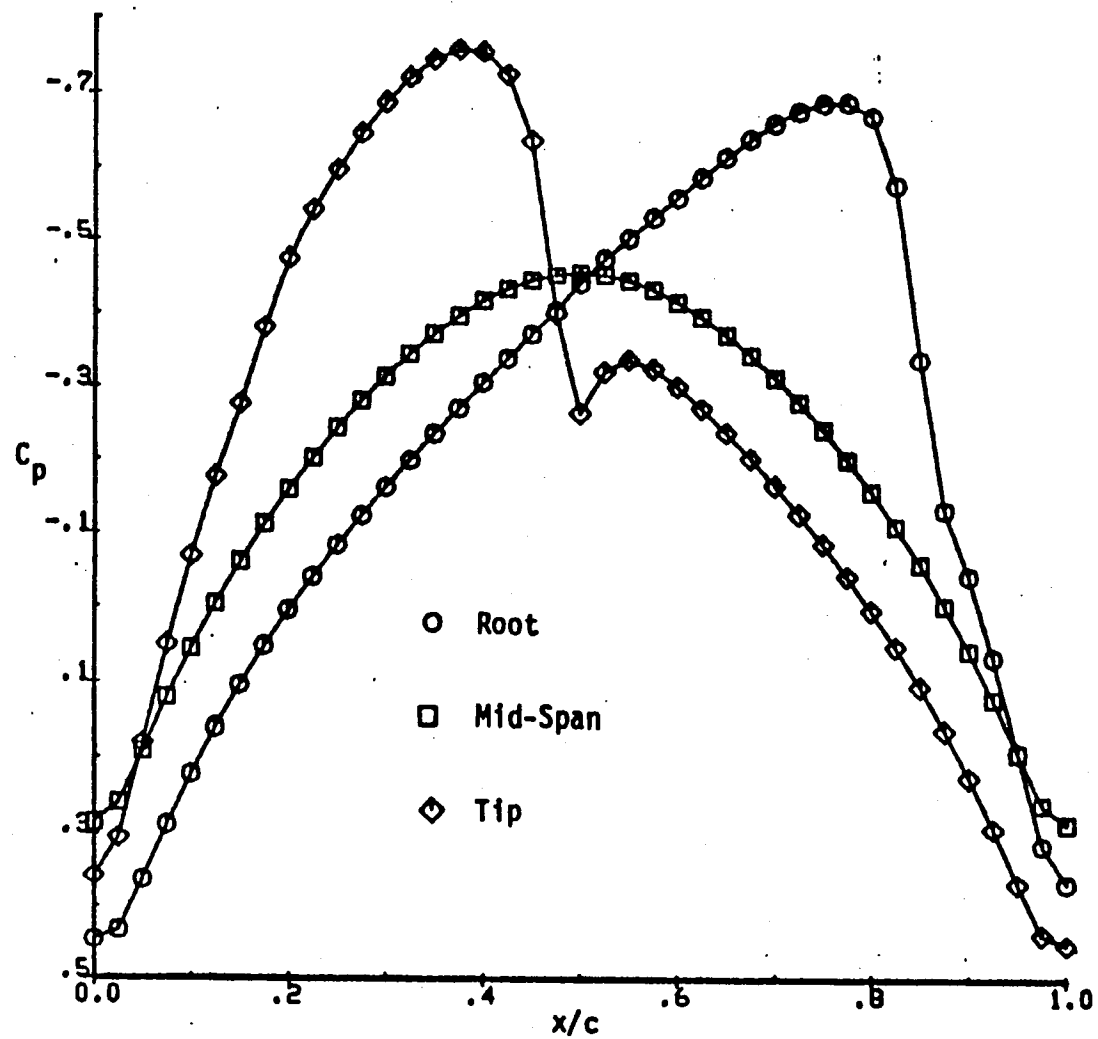


Figure 49. Surface  $C_p$  Distribution for Flow Over a Swept Wing With a .12 Parabolic Arc Wing Section, Planform-A, Grid-H,  $M_\infty = 0.8$ ,  $AR = 9.5$ .

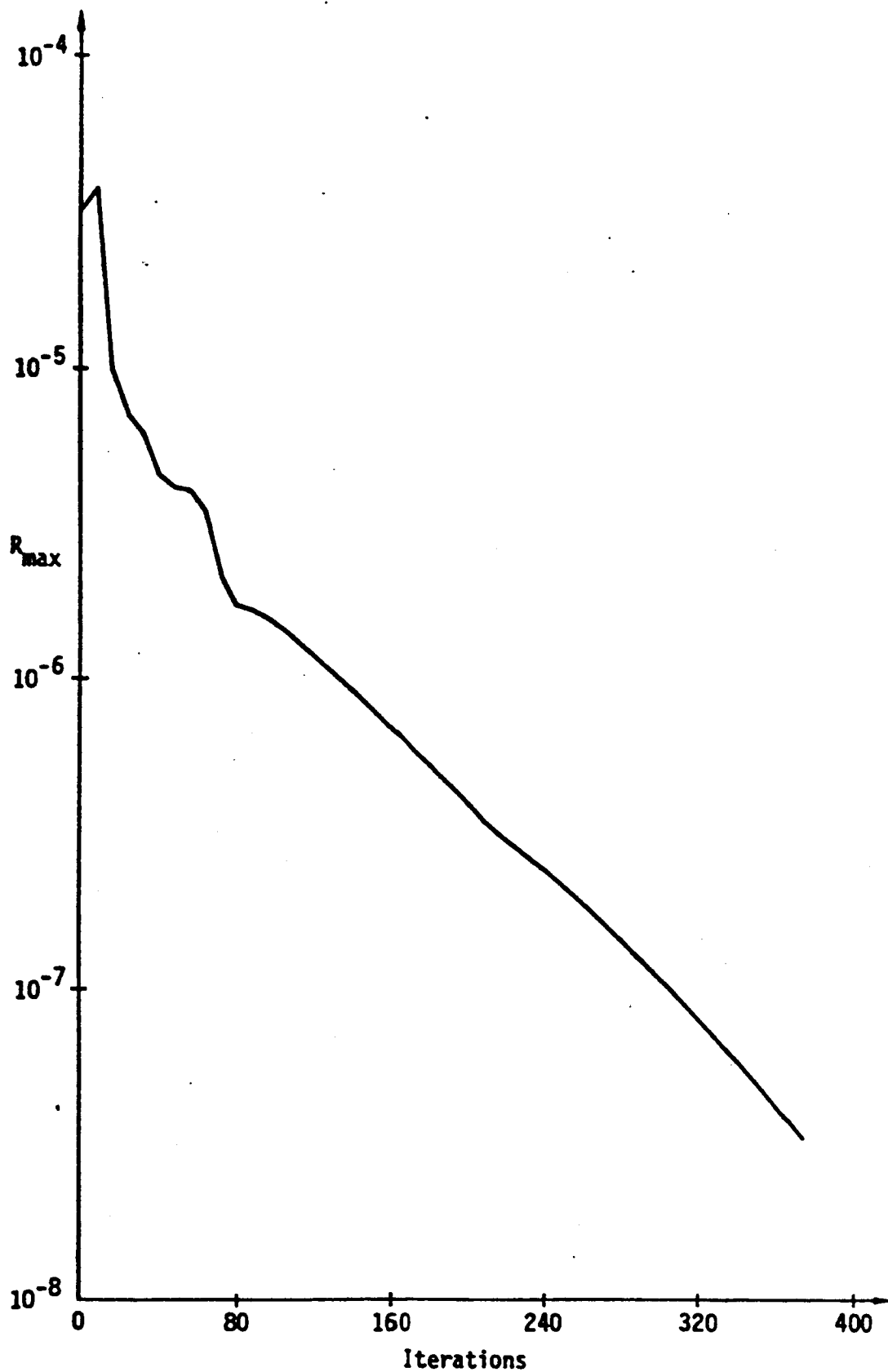


Figure 50a. Convergence History of  $R_{\max}$  for Flow Over a Swept Wing With a .12 Parabolic Arc Wing Section, Planform-B, Grid-H,  $M_{\infty}=0.8$ ,  $Ar=1.9$ .

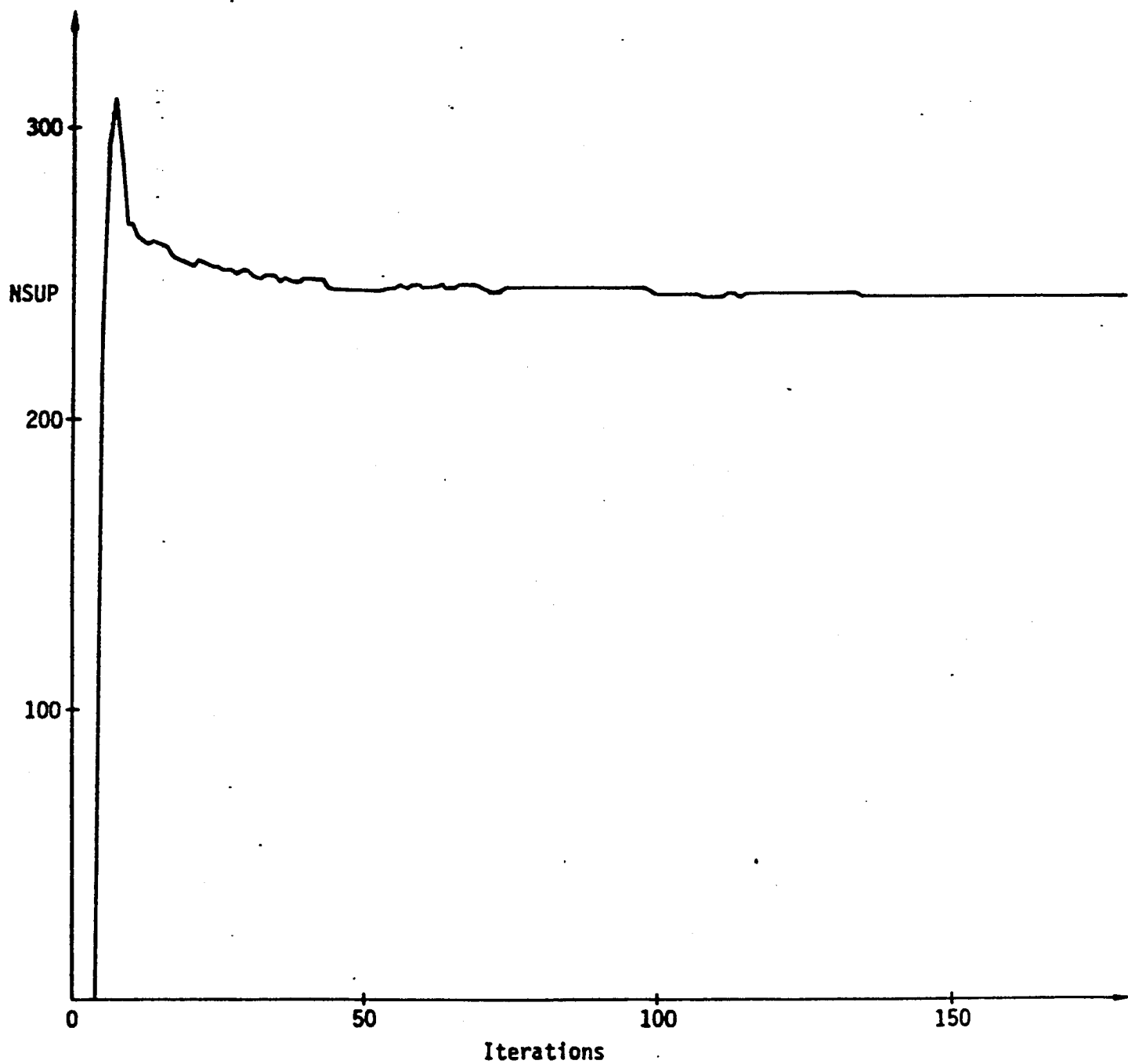


Figure 50b. Convergence History of NSUP for Flow Over a Swept Wing With a .12 Parabolic Arc Wing Section, Planform-B, Grid-H,  $M_\infty=.8$ ,  $AR=1.9$ .

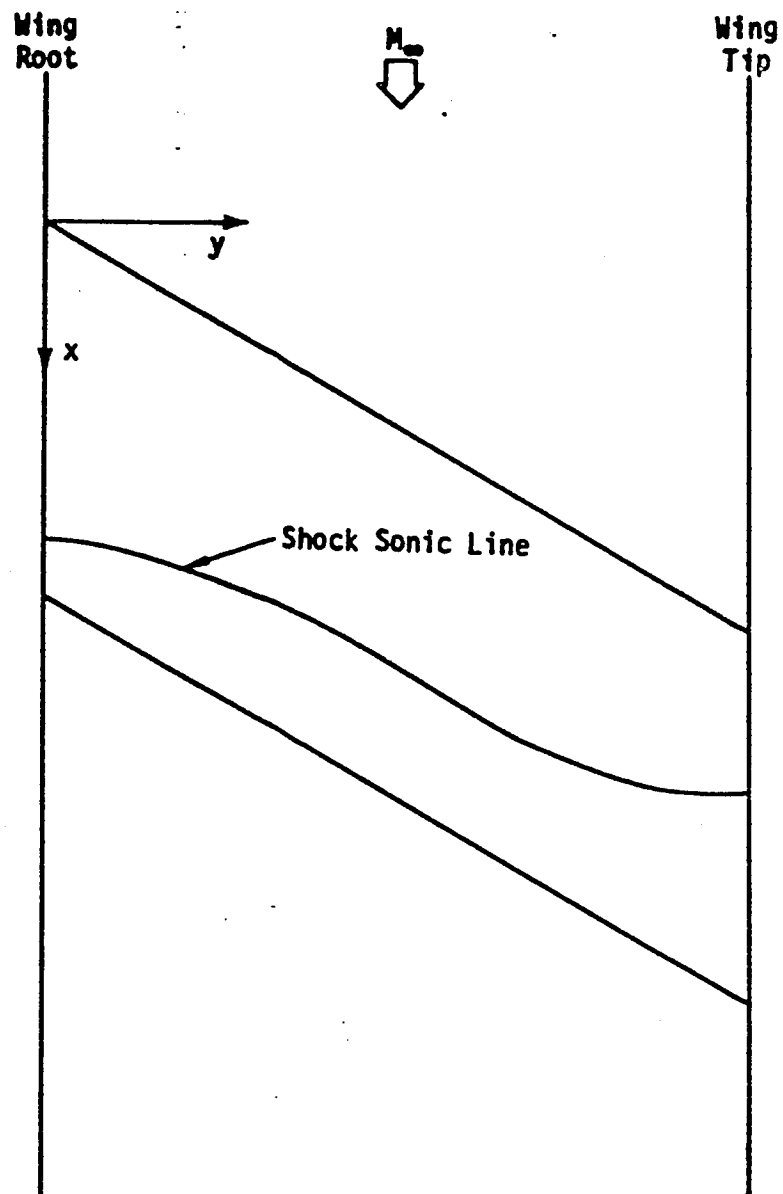


Figure 51. Shock Wave Position Relative to the Wing Planform, Planform-B, AR=1.9.

and trailing edge. This is qualitatively the same as the high aspect ratio wing case. The surface  $C_p$  distribution in the root, center span and tip plans are given in Figure 52.

#### C. Test Case Study Summary

Results of this chapter fully back up the grid effect study on TWINGB code given in Chapter IV where a method to improve the TWINGB flow solution algorithm is also presented. It is necessary that this improved method be adapted for extending TWINGB more effectively to flow problems involving arbitrary grids with block structures.

A summary of TWINGB test case study is given in Table 3.

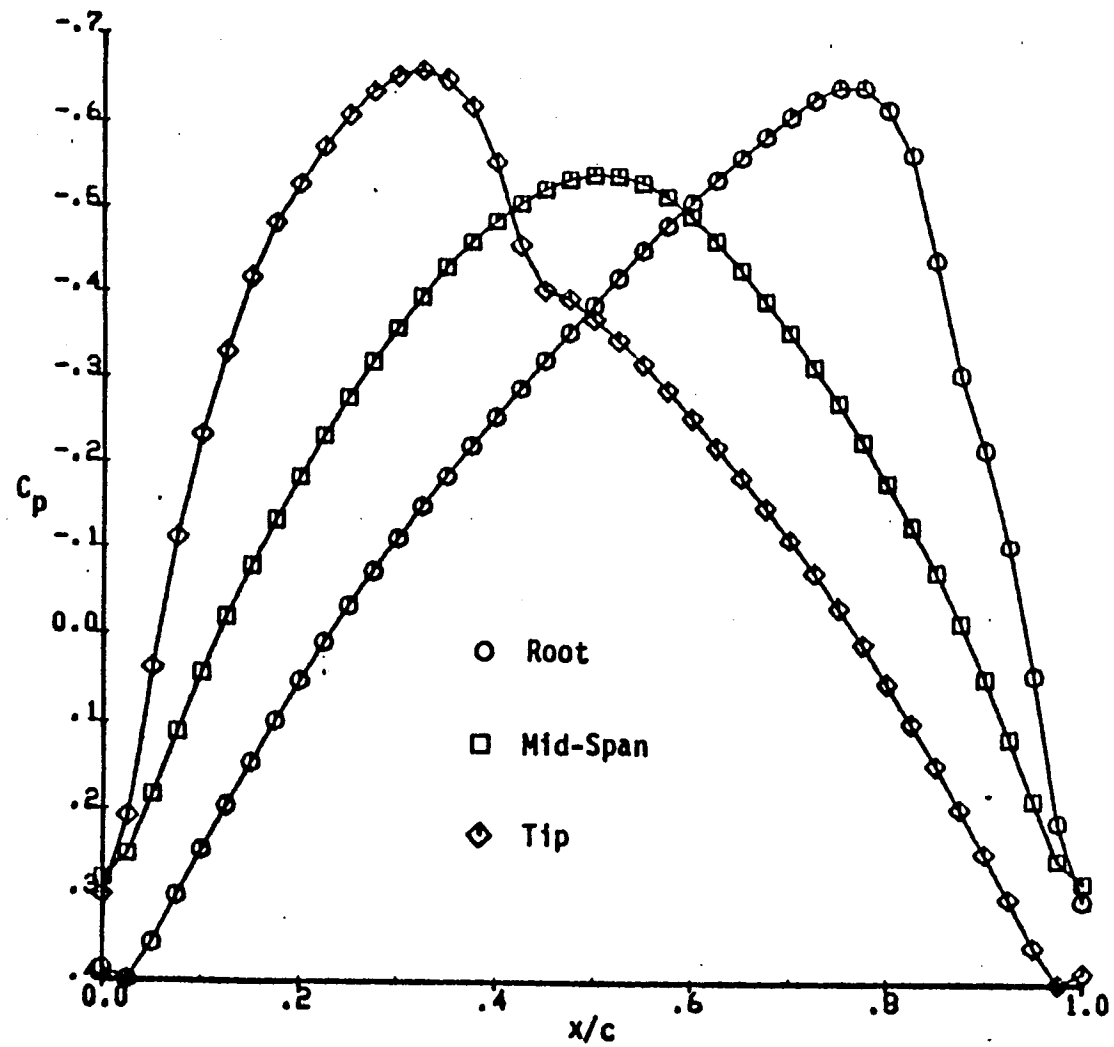


Figure 52. Surface  $C_p$  Distributions for Flow Over a Swept Wing With a .12 Parabolic Arc Wing Section, Planform-B, Grid-H,  $M_\infty=.8$ ,  $AR=1.9$ .



Table 3 TWINGB Case Study Summary

Airfoil Problem				
Thickness Ratio	M	Grid	Convergence in $R_{\max}$ to $10^{-3}$	Convergence in NSUP
0.03	0.75	Grid-E	37	
0.12	0.80	Grid-F	Diverge	
0.12	0.0001	Grid-F	Converging, terminated at 50 iterations	
0.12	0.80	Grid-G	Diverge	
0.12	0.80	Grid-H	119	27
30° Swept Wing AR=9.5				
Thickness Ratio	M	Grid	Convergence in $R_{\max}$	Convergence in NSUP
0.03	0.75	Grid-E	103	
0.03	0.75	Grid-I	54	
0.12	0.80	Grid-H	123	53
30° Swept Wing AR=1.9				
Thickness Ratio	M	Grid	Convergence in $R_{\max}$	Convergence in NSUP
0.03	0.75	Grid-I	274	
0.12	0.80	Grid-H	373	135

## VII. CONCLUSIONS

This study explores means for extending AF schemes to the treatment of transonic flows requiring the use of complex, multi-block grids. A method involving "zonal decomposition" with "interfacing grids" was developed. Test case studies for incompressible flow problems are completed and successful. This method is judged to be feasible for extension to transonic flows wherein the block boundaries are everywhere subcritical. The case of mixed flow along a block boundary will be much more difficult..

This study also adapted a fully implicit AF based 3D transonic code - called TWINGB to a block structured grid. Subsequent numerical experiments on flow over parabolic arc airfoils and wings using block-structured grids showed that the flow solution algorithm in TWINGB is sensitive to grid effects. The present conclusion is that the usefulness of TWINGB in the present form is limited to problems having only high quality grids. Uneven grid spacing or high aspect ratio grid cells lead to difficulties. An improved method to reduce the grid dependency of the flow solution algorithm in TWINGB is presented. The analysis and test case results in this study suggest that it is necessary to adapt the improved method for extending TWINGB more effectively to flow problem involving arbitrary grids with block structure.

## VIII. RECOMMENDATIONS

The experiences of this study results in the following four recommendations:

A. Compatibility between flow solution algorithm and computational grids must be carefully studied. This has far-reaching importance when developing computational transonic aerodynamic methods for flow over realistic airplane configurations using arbitrary grids.

B. Many approximation procedures in TWINGB are acceptable only for grids with very good quality (e.g., globally smoothly varying, moderate grid aspect ratio). To extend TWINGB to complex configurations with more general grids, it is necessary that the approximation procedures in TWINGB be improved. Specifically, the mapping procedure between the physical space and the computational space must be fully consistent.

C. Block-structured or segmented grids offer a sensible way to discretize the computational domain. Within each block, a grid of very good quality can be generated. The assemblage of these blocks provides the complete grid for computation. Generally, this leads to nonanalytical grid block boundaries and nonstandard grid cells. A logical treatment for this class of grids is to solve the flow by zonal decomposition using interfacing grids. In so doing, the problems with the nonanalytical grid block boundaries and nonstandard grid cells are circumvented by imposing suitable inter-block continuity conditions between neighboring blocks of grid in the flow solution algorithm. The success of applying the zonal decomposition method on the model problem in Chapter IV warrants a further study of this method for transonic wing problems.

D. Refinement of the multi-block grid generation is recommended. Specifically, linear equations for block grid generation should be replaced by nonlinear ones. This would ease the control of interior grid points in each block and therefore produce a grid with a desirable inter-block continuity of the grid points. The effect due to this improvement is expected to be significant when block boundaries are highly curved.

## IX. REFERENCES

1. Holst, T. L., 1979, "A Fast, Conservative Algorithm for Solving the Transonic Full-Potential Equation", AIAA, Vol. 18, 1980, pp. 1431-1439.
2. Lee, K. D., "3-D Transonic Flow Computations Using Grid Systems with Block Structure", AIAA, Fifth Computational Fluid Dynamic Conference, Palo Alto, California, June 22-23, 1981.
3. Lee, K. D., and Rubbert, P. E., "Transonic Flow Computations Using Grid Systems with Block Structure", Proceedings, the 7th International Conference on Numerical Methods in Fluid Dynamics, Stanford University, California, June, 1980.
4. Thompson, J. F., Thames, F. C., and Mastin, C. M., "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Arbitrary Two Dimensional Bodies", Journal of Computational Physics, Vol. 15, 1974, pp 299-319.
5. Yu, N. J., "Grid Generation and Transonic Flow Calculations for Three-Dimensional Configurations", AIAA Paper 80-1391, AIAA 13th Fluid and Plasma Dynamics Conference, Snowmass, Colorado, July, 1980.
6. Middlecoff, J. F. and Thomas, P. D., "Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations", AIAA Paper 79-1462, 1979.
7. Jameson, A., "Iterative Solution of Transonic Flows Over Airfoils and Wings, Including Flows at Mach 1", Communications on Pure and Applied Mathematics, Vol. 27, 1974, pp 283-309.
8. Holst, T. L., "An Implicit Algorithm for the Conservative Transonic Full Potential Equation Using an Arbitrary Mesh", AIAA Paper 78-1113, July, 1978.
9. Holst, T. L. and Ballhaus, W. F., "Fast Conservative Schemes for the Full Potential Equation Applies to Transonic Flows", AIAA, Vol. 17, 1979, pp. 145-152.
10. Ballhaus, W.F., Jameson, A., and Albert, T., "Implicit Approximate-Factorization Schemes for the Efficient Solution of Steady Transonic Flow Problems", AIAA J., Vol. 16, 1978, pp. 573-579.

## APPENDIX A

### CONSISTANT MAPPING AND VELOCITY CALCULATION

**Theorem.** Given a velocity potential for a uniform flow, in order to produce a uniform velocity, it is necessary and sufficient that the mapping procedure be fully consistent.

#### Necessary Proof

Let  $x, y, z$  be the physical coordinates and  $\xi, \eta, \zeta$  be the computational coordinates and the mapping between the physical and computational spaces is one-to-one, we have

$$\begin{Bmatrix} \phi_{\xi} \\ \phi_{\eta} \\ \phi_{\zeta} \end{Bmatrix} = \begin{Bmatrix} x_{\xi} & y_{\xi} & z_{\xi} \\ x_{\eta} & y_{\eta} & z_{\eta} \\ x_{\zeta} & y_{\zeta} & z_{\zeta} \end{Bmatrix} \begin{Bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{Bmatrix} \quad (A1)$$

In equation (A1), the  $\xi, \eta$ , and  $\zeta$  differentiations of  $\phi, x, y$ , and  $z$  are usually approximated by difference formulas in the  $\xi, \eta, \zeta$  coordinates. We thus approximate equation (A1) by equation (A2)

$$\begin{Bmatrix} \phi_{\xi} \\ \phi_{\eta} \\ \phi_{\zeta} \end{Bmatrix}_{\phi} = \begin{Bmatrix} x_{\xi} & y_{\xi} & z_{\xi} \\ x_{\eta} & y_{\eta} & z_{\eta} \\ x_{\zeta} & y_{\zeta} & z_{\zeta} \end{Bmatrix}_{xyz} \begin{Bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{Bmatrix} \quad (A2)$$

where subscript  $\phi$  denotes that the  $\phi$  derivatives in  $\xi, \eta$  and  $\zeta$  are approximated by difference formulas for  $\phi$ , and subscript  $xyz$  denotes that the  $x, y$ , and  $z$  derivatives in  $\xi, \eta$  and  $\zeta$  be approximated by difference formulas for  $x, y$ , and  $z$ .

If  $\phi = x$ , then

$$\phi_x = 1, \quad \phi_y = \phi_z = 0 \quad (A3)$$

and

$$\left\{ \begin{matrix} \phi_{\xi} \\ \phi_{\eta} \\ \phi_{\zeta} \end{matrix} \right\}_{\phi} = \left\{ \begin{matrix} x_{\xi} \\ x_{\eta} \\ x_{\zeta} \end{matrix} \right\}_{\phi} \quad (A4)$$

Also, from equations (A2) and (A3), if we indeed produce a uniform flow, then

$$\left\{ \begin{matrix} \phi_{\xi} \\ \phi_{\eta} \\ \phi_{\zeta} \end{matrix} \right\}_{\phi} = \left\{ \begin{matrix} x_{\xi} & y_{\xi} & z_{\xi} \\ x_{\eta} & y_{\eta} & z_{\eta} \\ x_{\zeta} & y_{\zeta} & z_{\zeta} \end{matrix} \right\}_{xyz} \left\{ \begin{matrix} 1 \\ 0 \\ 0 \end{matrix} \right\} = \left\{ \begin{matrix} x_{\xi} \\ x_{\eta} \\ x_{\zeta} \end{matrix} \right\}_{xyz} \quad (A5)$$

From equations (A4) and (A5)

$$\left\{ \begin{matrix} x_{\xi} \\ x_{\eta} \\ x_{\zeta} \end{matrix} \right\}_{\phi} = \left\{ \begin{matrix} x_{\xi} \\ x_{\eta} \\ x_{\zeta} \end{matrix} \right\}_{xyz} \quad (A5)$$

Thus the difference formula for  $\phi$  and  $x$  must be consistent. Similarly, we can set  $\phi = y$  and  $\phi = z$  to show that the difference formulas for  $\phi$  and  $x, y, z$  must be consistent.

Sufficient Proof

Assuming the difference formulas for  $\phi$  and  $x, y, z$  are inconsistent, we may replace the subscripts  $\phi$  and  $xyz$  by a subscript  $D$ . We then rewrite equation (A2) as

$$\begin{Bmatrix} x_{\xi} & y_{\xi} & z_{\xi} \\ x_n & y_n & z_n \\ x_{\zeta} & y_{\zeta} & z_{\zeta} \end{Bmatrix}_D \begin{Bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{Bmatrix} = \begin{Bmatrix} \phi_{\xi} \\ \phi_n \\ \phi_{\zeta} \end{Bmatrix}_D \quad (A7)$$

if  $\phi = x$ , then

$$\begin{Bmatrix} x_{\xi} & y_{\xi} & z_{\xi} \\ x_n & y_n & z_n \\ x_{\zeta} & y_{\zeta} & z_{\zeta} \end{Bmatrix}_D \begin{Bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{Bmatrix} = \begin{Bmatrix} x_{\xi} \\ x_n \\ x_{\zeta} \end{Bmatrix}_D \quad (A8)$$

If the mapping matrix is nonsingular, then equation (A8) has a unique solution

$$\begin{Bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} \quad (A9)$$

This is the uniform flow sought.

1. Report No. NASA CR-166316	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle A Development of Grid Generation Procedure for Multicomponent Aerodynamic Configuration		5. Report Date November 1981	
		6. Performing Organization Code	
7. Author(s) H. C. Chen		8. Performing Organization Report No.	
		10. Work Unit No.	
9. Performing Organization Name and Address The Boeing Company P.O. Box 3707 Seattle, WA 98124		11. Contract or Grant No. NAS2-10676	
		13. Type of Report and Period Covered Contractor Report	
12. Sponsoring Agency Name and Address National Aeronautics & Space Administration Washington, DC 20546		14. Sponsoring Agency Code	
15. Supplementary Notes ARC Technical Monitor: Dochan Kwak, 202A-14, Applied Computational Aerodynamics Branch, NASA Ames Research Center, Moffett Field, CA 94035 (415) 965-6415 FTS 448-6415			
16. Abstract  An extension of the approximation factorization (AF) scheme to multi-block grids is studied. An innovative method involving zonal decomposition using interfacing grids is investigated. Suitably matching the solutions at a block boundary for an incompressible flow problem it is demonstrates that this method is feasible for both contiguous and noncontiguous interfacing grids. Remaining issues for extending this method to transonic flow problem are discussed. The flow solution algorithm of a fully implicit AF code (TWINGB) is investigated for adaptation to block-structured grids. The analysis shows that the TWINGB code is sensitive to grid effects. A method to improve the TWINGB flow solution algorithm is presented to reduce the grid dependency of the code. Test case studies for subsonic and transonic flow over parabolic arc airfoils and wings suggest that the improved method should be adapted for extending TWINGB more effectively to flow problem involving arbitrary grids with block structure.			
17. Key Words (Suggested by Author(s)) Block Structured Grid Generations Approximation Factorization Schemes Transonic Computational Aerodynamics Zonal Decomposition		18. Distribution Statement  Unclassified - Unlimited  Subject Category - 01	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 105	22. Price*



**End of Document**